Міністерство освіти і науки України

Харківський національний університет ім. В. Н. Каразіна

Факультет комп’ютерних наук

Спеціальність 125 «Кібербезпека»

Освітня програма «Безпека інформаційних та комунікаційних систем»

«Допущено до захисту»

Зав.кафедрою БІСТ

Ольга Мелкозьорова

_____

« » 2023 р.

Пояснювальна записка

до кваліфікаційної роботи магістра

на тему: «Рішення проблеми масштабованості блокчейну за допомогою технології ZK-SNARK»

оцінка « »

Голова ЕК

Олександр ЛЕМЕШКО _____

Керівник: доктор технічних наук,
професор кафедри ЕтаУС
Краснобаєв В.А.

Консультант: співзасновник Zpoken.io
Єжов А.І.

Рецензент: доктор технічних наук,
професор кафедри ТПС
Толстолузька О. Г.

Виконавець: студентка групи КБ-61
Кузнецова К.О.

Харків – 2023

РЕФЕРАТ

Роботу присвячено викладенню основних концепцій технології блокчейн та опису ключових параметрів роботи блокчейн-технології для викладення проблеми масштабованості блокчейн мереж та аналізу її особливостей, вивчення існуючих напрямів вирішення масштабованості блокчейн, аналіз та порівняння відомих протоколів.

Для детального вивчення було обрано технологію доказів з нульовим знанням, на основі протоколів якої розроблено систему перевірки валідності ланцюга блоків. Наведені експериментальні дослідження обґрунтовують перспективність даного напрямку для вирішення проблем масштабованості сучасних блокчейн систем.

Актуальність обраної теми зумовлена необхідністю впровадження блокчейн систем в різні галузі людського життя. Однак, із розвитком будь-якої мережі зростає об’єм інформації, що необхідно безперервно обробляти. Цей виклик змушує розробляти рішення для вдосконалення систем, роблячи їх гнучкими у роботі з мільйонами користувачів. Водночас вкрай важливим питанням є підтримка безпеки та конфіденційності даних в оновлених системах та дотримання децентралізованої організації процесу обміну даними.

Отже, у сучасному світі блокчейн індустрії головним питанням є пошук моделей та методів для вирішення проблеми масштабованості мереж для подолання бар’єру повномасштабного впровадження блокчейн додатків.

Пояснювальна записка містить 80 сторінок, 12 рисунків, 16 таблиць та 83 джерел.

Об'єкт дослідження – процес блокчейн технології.

Мета роботи – дослідження проблеми масштабованості блокчейн.

Метод дослідження – теоретична та експериментальна програмна реалізація системи перевірки валідності ланцюга блоків із застосуванням технології ZK-SNARK.

Запроваджена система перевірки має можливість замінити традиційні методи перевірки, що призведе до значного прискорення перевірки валідності блоків блокчейн та підвищення зручності технології.

Ключові слова: БЛОКЧЕЙН, ТРИЛЕМА БЛОКЧЕЙН, ПРОБЛЕМА МАСШТАБОВАНОСТІ БЛОКЧЕЙН, ДОКАЗИ З НУЛЬОВИМ ЗНАННЯМ, ZK-SNARK, PLONK, FRI.

ABSTRACT

The work describes the main concepts of blockchain technology and its key parameters to present the scalability problem of the modern blockchain networks and analyze its features. It also studies existing directions and compares well-known protocols to propose the solution for the blockchain scalability problem.

We chose zero-knowledge proof technology as a promising direction for detailed study. We used protocols, based on this technology, to develop a validation system for a linked chain of blocks. Presented experimental results substantiate the prospects of this direction for solving the scalability problems of modern blockchain systems.

The relevance of the chosen topic is determined by the mass introduction of blockchain systems in various areas of human life. As it happens to every network, the volume of information that must be continuously processed increases. This challenge demands to develop solutions to improve systems, making them flexible in working with millions of users. At the same time, it is still important to maintain the security and confidentiality of the information and keep the decentralized organization of the data exchange process in the updated systems.

Therefore, in today's world of blockchain industry, the main issue is to find models and methods to solve the problem of scalability of networks to overcome the barrier of full-scale implementation of blockchain applications.

The work consists of 80 pages, 12 figures, 16 tables and 83 sources.

The object of research is the process of blockchain technology.

The purpose of the work is to study the problem of blockchain scalability.

The research method is a theoretical and experimental software implementation of the validation system for a linked chain of blocks using ZK-SNARK technology.

The introduced verification system has the ability to replace the traditional one, which may lead to a significant acceleration of the verification of data validity in blockchain and increase the convenience of the technology.

Keywords: BLOCKCHAIN, BLOCKCHAIN TRILEMMA, BLOCKCHAIN SCALABILITY PROBLEM, ZERO-KNOWLEDGE PROOFS, ZK-SNARK, PLONK, FRI.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

ОЦ	– обчислювальна цілісність
ЦП	– цифровий підпис
QAP	– програма квадратичної арифметики
PCS	– схема поліноміального зобов’язання
MPC	– багатостороннє обчислення
KZG	– схема поліноміального зобов'язання (названа на честь його початкових винахідників Кейт, Заверухи та Голдфедера)
DLT	– Distributed Ledger Technology
P2P	– Peer-to-Peer
BTC	– Bitcoin
ZKP	– Zero-Knowledge proofs
ZK-SNARK	– Zero-Knowledge Succinct Non-Interactive Argument of Knowledge
ZK-STARK	– Zero-Knowledge Scalable Transparent Argument of Knowledge
PLONK	– Permutations over Lagrange-bases for Oecumenical Non-interactive arguments of Knowledge
FRI	– Fast Reed-Solomon Interactive Oracle Proof
CRS	– Common Reference String
SRS	– Structured Reference String
NFT	– Non-fungible token
SegWit	– Segregated Witness
PoW	– Proof of Work
PoS	– Proof of Stake

PoK – Proof of Knowledge

DPoS – Delegated Proof of Stake

KYC – Know your Client

HTTPS – HyperText Transfer Protocol Secure

ВСТУП

Блокчейн, революційна технологія розподіленого реєстру, що лежить в основі криптовалют, обіцяє трансформувати індустрії завдяки своєму незмінному, прозорому та децентралізованому механізму запису транзакцій. Проте внутрішня проблема масштабованості в технології блокчейн створює суттєвий бар'єр, що перешкоджає її широкому впровадженню.

Основна мета цього дослідження полягає в тому, щоб запропонувати перспективний метод вирішення проблеми масштабованості, властивої технології блокчейн. Це запропоноване рішення має бути універсальним і застосовним у різних системах. Щоб досягти цього, було проведено широкий огляд технології блокчейн, зосереджуючись особливо на проблемі масштабованості. Аналіз розбирає ключові особливості блокчейн, щоб відокремити найбільш релевантні напрямки для вдосконалення систем блокчейн. Крім того, надається пояснення до відомих напрямків, аналізуючи їх переваги та висвітлюючи проблеми та ризики, з якими вони стикаються.

Обраним напрямком для поглибленого вивчення є стислі неінтерактивні аргументи знання з нульовим знанням (ZK-SNARKs), які представляють багатообіцяючий шлях для досягнення надійної масштабованості, необхідної для широкого впровадження блокчейн. Було запроваджено та оцінено численні схеми для побудови доказів обчислювальної цілісності, включаючи рекурсивну генерацію доказів і процеси перевірки.

Використовуючи мову програмування Rust і протоколи PLONK та FRI у рамках Plonky2, було розроблено обчислювальні моделі та оцінено їх продуктивність як з точки зору обчислювальної складності, так і ефективності. Експериментальний аналіз охоплює сценарії, що включають автономні та зведені докази для окремих і кількох блоків даних. Результати висвітлюють компроміс між складністю створення доказів і швидкістю перевірки, підкреслюючи потенційні переваги рекурсивних доказів.

Це всебічне дослідження має за мету внести цінну інформацію в поточний дискурс щодо масштабованості блокчейн, прокладаючи шлях до більш масштабованих і ефективних систем блокчейн.

1 АНАЛІЗ ТА ДОСЛІДЖЕННЯ СУЧАСНИХ ТЕНДЕНЦІЙ ТЕХНОЛОГІЇ БЛОКЧЕЙН

Блокчейн представляє стійкий до втручання цифровий реєстр без центрального сховища та зазвичай без центрального органу прийняття рішень, який реалізується за допомогою зв'язування інформації в безперервний ланцюжок блоків.

На базовому рівні блокчейн дозволяє деякій спільноті користувачів реєструвати транзакції в спільному реєстрі в межах даної спільноти таким чином, що за нормальної роботи мережі жодна транзакція не може бути змінена після публікації [1].

1.1 Дослідження історичної ретроспективи виникнення технології блокчейн

Хоча фундаментальні концепції технології блокчейн, включаючи системи розподіленого реєста та криптографічні методи, мають більш тривалу історію, деякі ключові ідеї та розробки з’явилися наприкінці 1980-х і на початку 1990-х років. Ці ідеї заклали основу для остаточного створення блокчейн-систем.

Зокрема, роботи дослідників Ральфа Меркла та Стюарта Хабера зробили значний внесок у галузі криптографії та цифрової безпеки. Основними напрямками їхньої роботи були [2-4]:

- Конструкція Меркле-Дамгорда, яка забезпечує метод взяття блоку даних деякого розміру та хешування його у вихідні дані фіксованого розміру. Це дозволило ефективно хешувати дані довільної довжини, що є основою для сучасних криптографічних хеш-функцій.
- Криптографічні хеш-функції. Наприкінці 1980-х Меркл і Хабер разом з іншими дослідниками працювали над розробкою та аналізом криптографічних хеш-функцій, визначили їх внесок у цифрову безпеку, включаючи їх використання для перевірки цілісності даних та цифрових підписів.

- Безпечний зв’язок. Меркл і Хабер сприяли розробці методів забезпечення конфіденційності та цілісності даних під час передачі та зберігання.
- Позначки часу та цифрові підписи. Робота Меркла та Хабера передбачала використання криптографічних методів для позначення часу цифрових документів, забезпечення їх автентичності та підтвердження їх існування в певний момент часу.

Внесок Ральфа Меркла та Стюарта Хабера наприкінці 1980-х років допоміг прокласти шлях до розвитку безпечного цифрового зв’язку, перевірки цілісності даних і криптографічних методів, які є фундаментальними для ширшої сфери цифрової безпеки та остаточної появи технологія блокчейн.

Важливим внеском стала ідея цифрових підписів, криптографічного методу автентифікації відправника повідомлення або документа. Цифрові підписи забезпечують спосіб перевірки автентичності та цілісності цифрових даних. Роботи таких дослідників, як Вітфілд Діффі та Мартін Хеллман, заклала основу технології цифрового підпису:

- У своїй фундаментальній статті 1976 року під назвою «Нові напрямки в криптографії» [5] («New Directions in Cryptography») Діффі та Хеллман представили концепцію криптографії з відкритим ключем. Вони запропонували систему, в якій два різні, але математично пов’язані ключі, один відкритий і один приватний, можуть використовуватися для безпечного зв’язку.
- Протокол обміну ключами Діффі-Хеллмана, представлений у тій же статті 1976 року, є фундаментальним алгоритмом безпечного обміну ключами по незахищеному каналу зв’язку. Це дозволяє двом сторонам створити спільний секретний ключ без попереднього спілкування чи спільного секрету. Цей протокол забезпечує безпечне спілкування в Інтернеті та використовується в різних протоколах шифрування, включаючи HTTPS.
- Окрім роботи над криптографією з відкритим ключем, Діффі та Хеллман зробили важливий внесок у технологію цифрового підпису. Цифрові підписи забезпечують засіб перевірки автентичності та цілісності

цифрових документів і повідомлень. Вони створюються за допомогою закритого ключа відправника та можуть бути перевірені за допомогою відповідного відкритого ключа [6].

Концепції та протоколи, представлені Діффі та Хеллманом, були включені в міжнародні стандарти та протоколи, що робить їх невід'ємною частиною глобальної криптографічної інфраструктури. Крім того, їх інновації заклали основу для розробки технології блокчейн, яка значною мірою покладається на криптографічні принципи для захисту транзакцій і забезпечення автентичності даних. Їх новаторська робота продовжує впливати та формувати сферу кібербезпеки та цифрової конфіденційності.

Стаття Стюарта Хабера та В. Скотта Сторнетти 1991 року під назвою «Як поставити позначку часу на цифровий документ» («How To Time-Stamp a Digital Document») є основоположною роботою в галузі криптографії та цифрової безпеки. У цьому документі представлено концепцію безпечного позначення часу з використанням криптографічних методів, критичного аспекту забезпечення цілісності та автентичності цифрових документів.

Головним внеском цієї роботи є розробка методу безпечного позначення часу цифрових документів. Автори визнали необхідність доведення існування цифрового документа в певний момент часу, вирішуючи проблеми, пов'язані з датуванням заднім числом, підробкою та суперечками щодо часу створення документа.

Ключові поняття та механізми використані в роботи:

- Хабер і Сторнетта використовували криптографічні хеш-функції як фундаментальний будівельний блок своєї системи міток часу. Цей вибір дозволив створити унікальний цифровий відбиток або хеш-значення для кожного документа, що служить безпечним представленням вмісту документа.
- Автори представили концепцію з'єднання міток часу разом, щоб сформувати хронологічний запис міток часу документів. Кожна нова мітка часу містить хеш-значення попередньої мітки часу. Цей механізм

ланцюжка створює структуру, захищену від несанкціонованого доступу, де будь-яка зміна документа або часових позначок у ланцюжку стає легко виявленою.

- Щоб забезпечити надійність позначок часу, Хабер і Сторнетта запропонували використовувати надійний орган із встановлення позначок часу. Ця сутність періодично збирає позначки часу від користувачів, перевіряє цілісність документів і створює сертифікати позначок часу. Ці сертифікати можна використовувати як юридичний доказ у випадках, коли мітки часу документа є вирішальними.

Внесок роботи Стюарта Хабера та В. Скотта Сторнетти мала тривалий вплив на цифрову безпеку, юридичну практику та управління даними. Не вводячи прямо термін «блокчейн», їх робота поділяє фундаментальні принципи з технологією блокчейн, адже останньою концепція об’єднання часових позначок і забезпечення незмінності даних має сильні паралелі в структурі. Блокчейн, який став широко відомим завдяки біткойн, використовує подібні принципи об’єднання блоків (даних) разом із використанням криптографічних хешів для підтримки безпечного та прозорого реєстру.

Сфера розподілених систем, яка займається проектуванням, впровадженням і керуванням мережевими комп’ютерними системами, які працюють разом для досягнення спільної мети, суттєво вплинула на розробку та проектування технології блокчейн.

Нижче наведено огляд ключових концепцій і робіт в галузі розподілених системах, які згодом вплинули на розробку блокчейн технології:

- Алгоритми консенсусу. Алгоритм Paxos [8], представлений Леслі Лемпортом у 1989 році, є основоположним консенсусним алгоритмом, який використовується в розподілених системах. Це згодом надихнуло на розробку механізмів консенсусу в блокчейн, таких, наприклад, як Byzantine Fault Tolerance (BFT).

Алгоритми розподіленого консенсусу, такі як Raft [9] і PBFT [10] також надихнули на розробку механізмів консенсусу, таких як PoW, PoS, що використовуються в блокчейн.

- Децентралізація. BitTorrent [11], розроблений Бремом Коеном, представив концепцію однорангового обміну файлами Peer-to-Peer (P2P). Блокчейн-мережі використовують ідею розподілених вузлів і зв'язку P2P для досягнення децентралізації.
- Відмовостійкість. Реплікація Raft і Viewstamped [12] – алгоритми розподіленого консенсусу, такі як Raft і Viewstamped Replication, спрямовані на відмовостійкість у розподілених системах, що є ключовим аспектом надійності мережі блокчейн.
- Одноранговi мережі. Napster і Gnutella – одноранговi мережі обміну файлами – продемонстрували концепцію децентралізованих мереж, що вплинуло на розвиток архітектури блокчейн Peer-to-Peer [13].

Поєднання елементів розподілених систем, криптографії та механізми консенсусу пізніше було застосовано до електронної готівки. Так, 31 жовтня 2008 року Сатоші Накамото оприлюднив білу книгу, в якій запропонував Біткойн [14].

Біткойн – це онлайн-система цифрових грошових коштів P2P, яка не потребує довіреної третьої сторони. У Bitcoin учасники володіють правами власності на віртуальні криптовалюти, які позначаються як біткойни (BTC). Користувачі генерують транзакції для передачі BTC та зберігають їх у загальнодоступному блокчейн реєстрі. Найменша доля цінності BTC, відома як сатоші, еквівалентна 0,00000001 BTC. Ідеї криптовалюти були описані в статті «Біткойн: одноранговa електронна готівковa системa» («Bitcoin: A Peer-to-Peer Electronic Cash System») [15].

Нарешті, 3 січня 2009 року було створено генезис блок-реєстру криптовалюти Біткойн, де було реалізовано перший ланцюг блоків, який надалі використано в якості основи для блокчейн технології.

1.2 Аналіз способів застосування технології блокчейн

Технологія блокчейн у поєднанні з іншими технологіями та комп'ютерними концепціями дозволила розробити чимало криптовалютних систем: Bitcoin, Ethereum [16], Solana [17], BNB [18]тощо. У таких криптовалютних системах передача цифрової інформації, яка представляє електронну готівку, відбувається в розподіленій системі. Юзери можуть цифровим підписом передавати свої права на інформацію іншим користувачам, а мережа публічно фіксує цю передачу, дозволяючи всім учасникам самостійно перевіряти дійсність транзакцій. Такі системи незалежно обслуговуються та управляються розподіленою групою учасників. Це, разом із криптографічними механізмами, робить блокчейн стійким до спроб пізніше змінити обліковий реєстр [1, 19].

Блокчейн також використовується для укладання смарт-контрактів. Смарт-контракти — це самовиконувані контракти з умовами угоди, записаними безпосередньо в код (наприклад Solidity для Ethereum), що автоматизує процеси та зменшує потребу в посередниках [19]. Фізична або юридична особа генерує операції та наслідки потенційних ситуацій, що виникають під час юридичних договірних відносин. Смарт-контракти можуть гарантувати, що юридичний контракт буде виконано належним чином або що платіж, здійснений наперед, буде повністю або частково повернуто у разі будь-якого збою або навмисного невиконання домовленостей. У результаті блокчейн, який підтримує смарт-контракти, забезпечує «автоматизоване правосуддя», пропонуючи користувачеві можливість програмувати в коді безліч можливих обставин разом із відповідними наслідками [20, 21].

Смарт-контракти зазвичай розгортаються в блокчейн-мережах, таких як Ethereum. Блокчейн забезпечує безпеку, прозорість і незмінність виконання контракту [19].

За своєю первісною метою смарт-контракти пропонували надійну платформу для виконання договірних відносин, відключених від

централізованих офіційних органів. Тому вони дають можливість реалізовувати голосування [22], збір коштів або комерційний обмін без втручання держави (або іншого офіційного органу чи реєстру).

Смарт-контракти мають потенціал для революції в різних галузях, роблячи процеси більш ефективними, прозорими та безпечними. Однак для їх широкого впровадження може знадобитися час.

Blockchain може надавати безпечні та децентралізовані рішення для перевірки особи, зменшуючи крадіжку особистих даних і шахрайство [20].

Традиційні системи керування ідентифікацією несуть такі ризики, як виток даних, крадіжка ідентифікаційних даних, а також фрагментована структура системи, що вимагає від користувачів керування кількома іменами і паролями. Блокчейн вирішує ці проблеми, забезпечуючи безпечний, незмінний та децентралізований реєстр для ідентифікаційних даних. Блокчейн пропонує потенціал для створення єдиного джерела правди для перевірки особи, зменшуючи потребу в посередниках, підвищуючи тим самим безпеку. Блокчейн можна використовувати для керування ідентифікацією, наприклад, для перевірки особи претендентів на роботу або оптимізації процесів «Знай свого клієнта» (KYC) у фінансовому секторі.

Іншим застосуванням блокчейн технології є NFT (non-fungible token) – незамінний токен [23]. NFT – це криптографічні токени, які представляють право власності або доказ автентичності унікального предмета чи витвору, як правило, мистецтва, музики, предметів колекціонування або віртуальної нерухомості.

На відміну від криптовалют, які є взаємозамінними та рівновартісними (наприклад, один біткойн завжди дорівнює одному біткойну та ними можна обмінюватись), NFT є унікальними та не підлягають обміну. Кожен NFT представляє право власності або доказ автентичності певного цифрового чи фізичного предмета [24]. Вони забезпечують прозорий і незмінний запис про право власності та історію транзакцій. Це особливо цінно у світі мистецтва, де походження та автентичність є вирішальними.

Таким чином, NFT можна використовувати для зберігання набагато складнішої та індивідуальної інформації. Державні документи, такі як свідоцтва про шлюб, рейтинги харчових продуктів і водійські права тощо можна токенізувати за допомогою NFT. У роздрібній торгівлі споживачі можуть використовувати NFT для перевірки легітимності предметів розкоші.

NFT купуються та продаються на ринках NFT, таких як OpenSea [25] та Rarible [26], де користувачі можуть знаходити, купувати та торгувати NFT.

Попит цифрових примітивів на право власності продовжує зростати, та ринок NFT отримує цінність великими темпами.

В останні роки привернуло значну увагу використання технології блокчейн в ігровій індустрії [27]. Блокчейн може принести різноманітні переваги іграм, включаючи право власності на внутрішньоігрові активи, дефіцитні предмети та підвищену безпеку.

Блокчейн дозволяє гравцям мати справжнє право власності на ігрові активи, такі як скіни, зброя чи персонажі. Ці активи можна купувати, продавати та торгувати безпечним і прозорим способом.

Крім того, блокчейн забезпечує взаємодію між різними іграми, дозволяючи гравцям використовувати свої активи на кількох ігрових платформах і екосистемах.

Завдяки використанню NFT розробники ігор можуть створювати доказово дефіцитні предмети або персонажів, що робить їх дуже цінними та колекційними. Ігри, засновані на блокчейн, часто мають економіку, керовану гравцями, де гравці можуть заробляти криптовалюту або інші винагороди за свою внутрішньоігрову діяльність [28].

Першою відомою грою, яка використовує технології блокчейн, була CryptoKitties, запущена Axiom Zen у листопаді 2017 року для персональних комп'ютерів [29]. Гравець купував NFT за криптовалюту Ethereum, кожна NFT складалася з віртуального вихованця, якого гравець міг схрещувати з іншими, щоб створити потомство з комбінованими рисами як нові NFT. Гра потрапила

до заголовків газет у грудні 2017 року, коли один віртуальний вихованець був проданий більш ніж за 100 000 доларів США.

Технологія Blockchain підвищує безпеку, запобігаючи шахрайству та несанкціонованому дублюванню внутрішньоігрових активів. Також гравці мають контроль над своїми активами, що зменшує ризик блокування облікових записів або конфіскації активів розробниками ігор.

Це лише кілька прикладів того, як технологія блокчейн застосовується в різних галузях. Потенційні застосування блокчейн продовжують розширюватися завдяки постійним інноваціям і експериментам. Кожен із цих варіантів використання пропонує потенціал для більшої прозорості, безпеки та ефективності у відповідних доменах.

1.3 Обгрунтування та аналітичний опис технології блокчейн

Технологія блокчейн – це децентралізована та розподілена система цифрового реєстру, який записує транзакції на кількох комп’ютерах таким чином, щоб забезпечити цілісність, безпеку та прозорість даних.

Головна задача, яку вирішує блокчейн – створення середовища без довірених посередників. Досягається це за допомогою наступних характеристик.

Технологія використовує *незмінний реєстр*, в який можна лише додавати дані [1]. На відміну від традиційних баз даних, транзакції та значення в блокчейн не перевизначаються.

Блокчейн зберігає записи всіх транзакцій у своїй мережі. Кожен учасник мережі має копію всього реєстру. *Розподіленість* дозволяє масштабувати кількість вузлів мережі блокчейн, щоб зробити її більш стійкою до атак зловмисників [30]. Збільшуючи кількість вузлів, зменшується здатність зловмисника впливати на протокол консенсусу, який використовується блокчейном.

Децентралізація є фундаментальною концепцією технології блокчейн, яка стосується розподілу контролю, повноважень і даних у мережі вузлів [31].

Блокчейн працює в мережі комп'ютерів, поширених по всьому світу, та не покладається на центральний орган влади чи посередника для перевірки та реєстрації транзакцій. Натомість транзакції перевіряються та додаються до блокчейн за консенсусом між учасниками мережі.

Публічна верифікація дозволяє будь-кому перевірити правильність стану системи [32]. У розподіленому реєстрі кожен крок зміни підтверджується групою верифікаторів. Користувач має можливість переконатися, що стан системи змінився відповідно до протоколу. Таким чином, всі учасники системи мають єдине уявлення про поточний стан системи щонайменше на певний період часу.

Ще одною характеристикою є *прозорість даних і процесу* оновлення стану, яка в свою чергу є вимогою для публічної перевірки [32]. Транзакції, зареєстровані в блокчейн, видимі всім учасникам мережі. Хоча особи зазвичай є псевдонімами (представлені криптографічними адресами), історія транзакцій відкрита для перевірки. Ця прозорість допомагає запобігти шахрайству та сприяє довірі користувачів.

Технологія блокчейн охоплює два суперечливі аспекти: прозорість та *конфіденційність*. У повністю прозорій системі інформація доступна для всіх, що потенційно може порушити конфіденційність. І навпаки, повна конфіденційність може призвести до відсутності довіри в мережі. У загальнодоступних блокчейн-системах конфіденційність підтримується за допомогою криптографічних методів, що забезпечує як анонімність, так і достатній ступінь прозорості для контролю стану реєстру.

Цілісність інформації гарантує, що інформація захищена від несанкціонованих змін, тобто, що отримані дані є правильними [33]. Механізми криптографічної безпеки Blockchain служать для збереження цілісності даних книги, запобігаючи будь-якому несанкціонованому втручанню. Це збереження цілісності даних тісно пов'язане з можливістю полегшити публічну перевірку. Коли система допускає публічну перевірку, це дає будь-кому можливість незалежно підтвердити цілісність даних.

Для забезпечення конфіденційності та цілісності оброблюваної в системі інформації використовуються криптографічні хеш-функції. Хешування, як техніка, включає застосування криптографічної хеш-функції до даних та обчислює відносно унікальний вихід – дайджест. Він створюється незалежно від розміру вхідних даних, гарантуючи, що будь-хто може окремо взяти вхідні дані, хешувати їх і отримати ідентичний результат. Цей процес є доказом того, що дані залишаються недоторканими та незмінними.

Криптографічні хеш-функції мають такі властивості безпеки [34]:

- Стійкі до прообразу: криптографічні хеш-функції односторонні, що означає, що обчислювально неможливо визначити дійсне вхідне значення, якщо йому надається конкретне вихідне значення. Іншими словами, якщо у вас є дайджест *h*, знайти *x*, щоб *hash(x) = h*, було обчислювально нездійсненним завданням.
- Стійкість до другого прообразу: хеш-функції також призначені для запобігання атакам другого прообразу, що робить обчислювально малоймовірним знайти альтернативний вхідний сигнал *y*, який дасть той самий результат, що й заданий вхідний сигнал x. Простіше кажучи, якщо задано *x*, ідентифікувати *y* так, що *hash(x) = hash(y)* є надзвичайно складним завданням та зазвичай потребує вичерпних пошуків у множині вхідних значень.
- Стійкість до зіткнень (колізій). Криптографічні хеш-функції додатково захищають від колізійних атак, коли обчислювально неможливо ідентифікувати будь-які два різні вхідні дані *x* і *y*, які мають один і той самий дайджест, тобто знайти *x* і *y* так, що *hash(x) = hash(y)* є надзвичайно складним обчислювальним завданням.

Існує багато сімейств криптографічних хеш-функцій, які використовуються в блокчейн: SHA-256 [35], Keccak [36], RIPEMD-160 [37] тощо.

Таким чином блокчейн захищає транзакції та контролює доступ до даних. Транзакції перевіряються за допомогою механізму консенсусу (наприклад, Proof

of Work або Proof of Stake), що ускладнює зловмисникам маніпулювання обліковою книгою.

Технологія забезпечує безпечний, прозорий та ефективний спосіб створення та перевірки транзакцій. Її застосування можливе в різних галузях, що робить блокчейн важливою сферою інтересів та інновацій в епоху цифрових технологій.

1.4 Дослідження трилеми блокчейн та проблеми масштабованості

Блокчейн – це децентралізований розподілений реєстр (DLT), що реалізується за допомогою зв'язування інформації в безперервний ланцюг блоків, схематично представлено на рисунку 1.1.

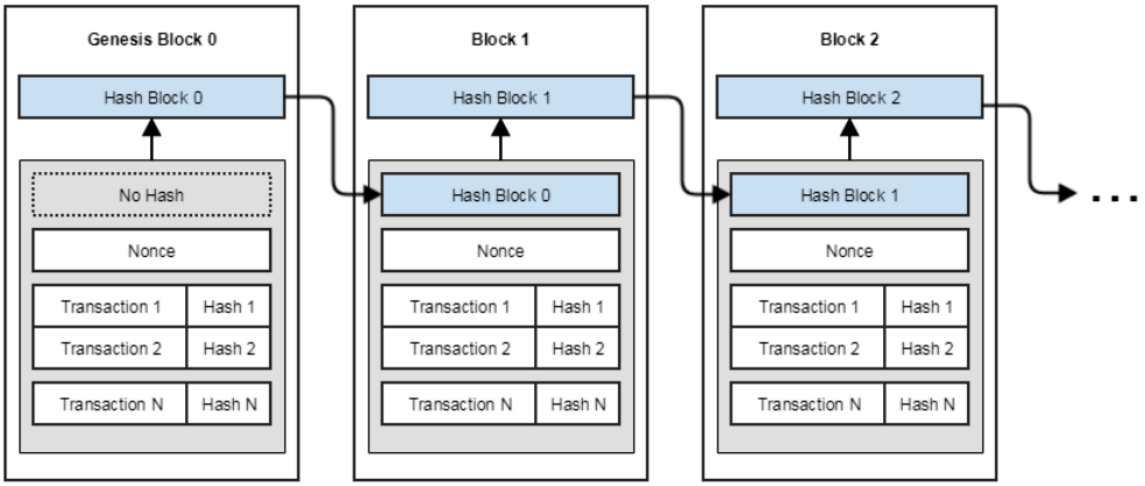

Рисунок 1.1 – Схематичне представлення ланцюга блоків

Зв'язок між блоками забезпечується криптографічним механізмом через обчислення хеш-суми попереднього блоку.

Система повинна забезпечувати достатній рівень безпеки та в певній мірі анонімності – зберігати право на приватне життя.

Крім безпеки система має дотримуватись децентралізації – не керуватись єдиним центром прийняття рішення, що вирішує питання надійності, адже централізована структура з потенційно єдиною точкою збою завжди приваблює зловмисників. Децентралізація захищає від цензурування, встановлює засади демократії прийнятих рішень, прояву свободи слова та незалежності думки.

До того ж, на послуги системи зростає попит, та вона має безперервно розвивається. Водночас, із збільшенням кількості користувачів та їх потреб, час обслуговування не повинен значно зростати. Це складна вимога, адже екстенсивно забезпечити масштабованість – тільки збільшенням числа комп'ютерів – не завжди вдається.

Таким чином, маємо три основні вимоги: безпека, децентралізація і масштабованість – взаємозв'язок яких сформульовано у відомій трилемі блокчейн, запропонованій Віталіком Бутеріним – співзасновником Ethereum, що привенено на рисунку 1.2 [38].

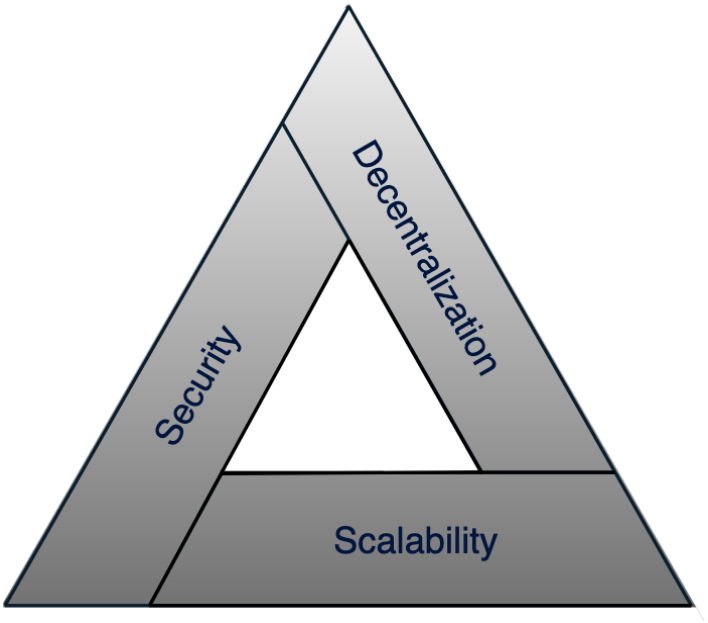

Рисунок 1.2 – Трилема блокчейн

Трилема блокчейн підкреслює труднощі досягнення всіх трьох цих властивостей одночасно. На практиці часто буває важко оптимізувати одну властивість без шкоди для інших. Так, з одного боку, підвищення безпеки може вимагати складних механізмів консенсусу або додаткових рівнів захисту, що потенційно може вплинути на масштабованість. З іншого боку, досягнення високої децентралізації може призвести до уповільнення часу підтвердження транзакцій і збільшення споживання ресурсів, що впливає на масштабованість. Водночас, покращення масштабованості може передбачати компроміси з точки

зору децентралізації, оскільки збільшення пропускної спроможності транзакцій у повністю децентралізованій мережі може бути складним завданням.

Наведемо декілька конкретних прикладів кожної сутності трилеми у таблиці додатку А.

Рішення, прийняті щодо трилеми блокчейн, мають значні наслідки для дизайну та продуктивності мережі блокчейн. Біткойн, наприклад, надає перевагу децентралізації та безпеці над масштабованістю, що призводить до меншого часу підтвердження транзакцій, що видщтовхує більшу частину користувачів. Сучасною практикою стало використання Біткойн як накопичувального рахунку, а не системою миттєвої оплати. Ethereum досліджував різні стратегії, включаючи перехід на Ethereum 2.0, щоб покращити масштабованість, зберігаючи при цьому певний рівень децентралізації та безпеки.

Зокрема, масштабованість впливає на дві інші характеристики, а саме: збільшення розміру мережі потенційно централізує контроль, адже великий обсяг даних отримуватиме нових користувачів до приєднання. Крім того, збільшення розмірів блоків, мають наслідки для безпеки, адже великі блоки можуть призвести до уповільнення розповсюдження даних у мережі, що, зрештою, потенційно полегшуючи майнерам маніпуляції блокчейном.

Отже, оскільки масштабованість потенційно може бути першопричиною нестабільності безпеки та децентралізації, то в даній роботі розглядається саме ця характеристика в її динаміці, оцінюється потенційні загрози для блокчейн систем, розглядаються існуючі підходи для оптимізації обсягів обчислень у мережі блокчейн та пропонується прискорений метод верифікації блоків мережі.

2 ВИВЧЕННЯ ПРОБЛЕМИ МАСШТАБОВАНОСТІ БЛОКЧЕЙН ТА АНАЛІЗ НАПРЯМКІВ ЇЇ ВИРІШЕННЯ

2.1 Вивчення проблеми масштабованості блокчейн

Проблема масштабованості в блокчейн є фундаментальною проблемою, яка перешкоджає здатності технології ефективно обробляти зростаючу кількість транзакцій, зберігаючи при цьому децентралізацію та безпеку. Ця проблема є основною проблемою для мереж блокчейн, особливо для публічних блокчейн, таких як Bitcoin та Ethereum.

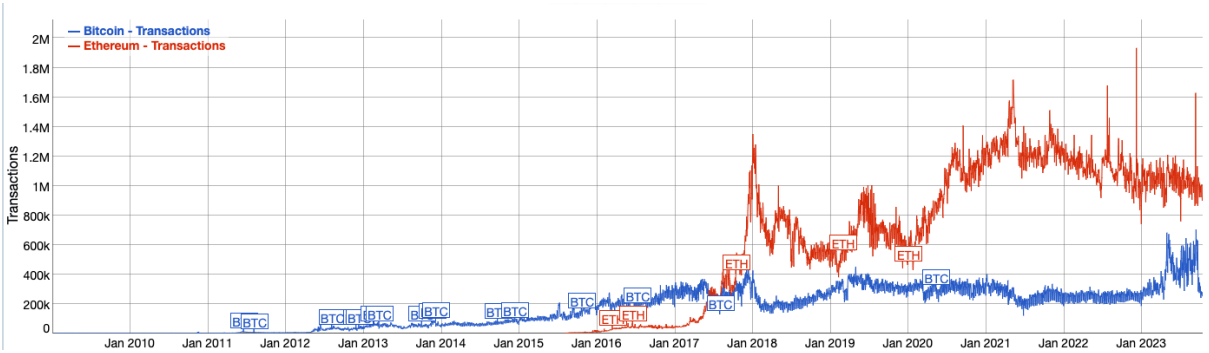

Рисунок 2.1 – Графік росту об’єму транзакцій Bitcoin та Ethereum [39]

Наведемо аналіз проблем, що виникають в системах із зростанням числа користувачів [40-42]:

1) Пропускна здатність транзакцій. Одним із основних аспектів масштабованості є пропускна здатність транзакцій, яка стосується кількості транзакцій, які блокчейн може обробити за секунду. Загальнодоступні блокчейн часто мають проблеми з обмеженою пропускною здатністю обробки транзакцій. Біткойн, наприклад, має обмеження на розмір блоку, яке обмежує кількість транзакцій, які можуть бути включені в блок.
2) Споживання ресурсів. Проблеми з масштабованістю можуть призвести до збільшення споживання ресурсів, що може зробити роботу мережі дорожчим.

3) Ресурсомісткість. Доказ роботи (PoW) і доказ частки володіння (PoS), найпоширеніші механізми консенсусу, мають обмеження щодо масштабованості. PoW, хоча й безпечний, може потребувати великих обчислень, тоді як PoS може важко обробляти великий обсяг транзакцій. Майнерам потрібна значна обчислювальна потужність, а споживання енергії може бути високим, що в майбутньому може призвести до централізації видобутку.
4) Збільшення розміру блокчейна. Зберігання повної копії блокчейн може бути складним для вузлів з обмеженою ємністю для зберігання. Це може стримувати нові вузли від приєднання до мережі, потенційно централізуючи контроль.
5) Час підтвердження. Час, необхідний для переміщення інформації через розподілену мережу, може спричинити затримки підтвердження транзакцій, особливо в глобальних мережах. У біткойн, наприклад, час підтвердження блоку може варіюватися від кількох хвилин до годин, що робить його непридатним для онлайн додатків.
6) Ризик централізації. Деякі рішення масштабованості можуть включати компроміси, які знижують рівень децентралізації. Наприклад, збільшення розмірів блоків для розміщення більшої кількості транзакцій може призвести до зменшення кількості вузлів, здатних перевіряти та зберігати весь блокчейн, що потенційно може централізувати контроль.
7) Ризик з точки зору безпеки. Збільшення розмірів блоків вносить сумніви щодо безпеки, оскільки великі за розміром блоки можуть перешкоджати швидкому поширенню даних у мережі, тим самим збільшуючи ймовірність того, що майнерам буде легше втручатися в блокчейн та перешкоджати роботі валідного ланцюга блоків.

Слід зазначити, що кожен великий проект DLT протягом свого розвитку стикається з обмеженням в обслуговуванні кількості користувачів, виділених ресурсів та часу. Кожному необхідно перевіряти довгі ланцюжки

криптографічних примітивів, чим довше працює проект, тим складніша й більш витратна стає перевірка.

Схема на рисунку 1.1 в повній мірі вирішує проблеми безпеки та децентралізації. Для зв’язування блоків і підтвердження їх автентичності використовуються криптографічні перетворення. Усі блоки схвалюються консенсусом учасників, тобто в системі відсутній головний вузол. Але перевірки записів у такому DLT вимагає чималих ресурсів. Наприклад, якщо ми візьмемо мільйонний блок, то необхідно обчислити мільйон хешів і мільйон підписів. Цей спосіб обчислювально важкий та вимагає великих часових витрат.

Найчастіше використовуваним методом перевірки валідності блоків є нативна перевірка реєстру [43]. Вона зазвичай відноситься до процесу валідації та перевірки, який виконує вузол під час отримання та обробки нового блоку транзакцій. Цей процес є фундаментальним аспектом безпеки блокчейн та гарантує, що всі транзакції в блоці є законними та відповідають правилам консенсусу мережі. Нативна перевірка включає кілька ключових компонентів:

- Правила консенсусу. Кожна мережа блокчейн має спеціальні правила консенсусу, які визначають дійсність блоку та його транзакцій. Наприклад, у біткойн ці правила включають перевірку дійсного підтвердження роботи, забезпечення того, щоб транзакції не перевищували ліміт розміру блоку, і перевірку цифрових підписів.
- Перевірка транзакцій. Нативна перевірка передбачає перевірку кожної окремої транзакції в блоці. Це включає перевірку цифрових підписів, забезпечення наявності у відправника необхідних коштів і підтвердження того, що транзакція відповідає правилам мережі (наприклад, відсутність подвійних витрат).
- Перевірка дерева Merkle. Структура дерева Merkle у блоці забезпечує ефективну перевірку. Підтверджуючи, що корінь Merkle у заголовку блоку збігається з обчисленим коренем Merkle усіх транзакцій, вузол може перевірити цілісність транзакцій у блоці без необхідності перевіряти кожну з них окремо.

Нативна перевірка є важливою частиною безпеки блокчейн, оскільки вона запобігає додаванню в блокчейн недійсних або зловмисних транзакцій. Однак це також сприяє проблемі масштабованості наступними способами [44]:

1) Час обробки. Виконання власних перевірок кожного блоку та його транзакцій може зайняти багато часу, особливо в мережах із великим обсягом транзакцій. Це призводить до довшого часу підтвердження та зменшення пропускної здатності транзакцій.
2) Обчислювальні ресурси. Для підтвердження та перевірки транзакцій, особливо в блокчейнах на основі PoW, таких як біткойн, потрібні значні обчислювальні ресурси. Цей ресурсомісткий процес впливає на загальну ефективність мережі.
3) Обмежена масштабованість. У міру зростання блокчейн та збільшення кількості транзакцій власна перевірка стає вузьким місцем, обмежуючи масштабованість мережі. Більша кількість транзакцій означає довший час підтвердження та вищі вимоги до ресурсів.

Таким чином, незважаючи на те, що нативна перевірка є важливою опорою безпеки блокчейн, вона загострює проблему масштабованості, подовжуючи час обробки та посилюючи вимоги до обчислювальних ресурсів. Цей складний баланс між забезпеченням цілісності блокчейн та досягненням ефективної масштабованості залишається головною проблемою для розробників і технологів блокчейн.

Проблема масштабованості в блокчейн є багатогранною проблемою, яка впливає на здатність технології пристосовувати зростаючу базу користувачів і більший обсяг транзакцій, зберігаючи децентралізацію та безпеку. Це вимагає тонкого балансу між наведеними трьома критичними факторами, і постійні дослідження та інновації мають вирішальне значення для пошуку ефективних рішень проблеми. Різні блокчейн-проекти використовують різні стратегії для подолання проблем масштабованості та підвищення життєздатності технології для широкого спектру застосувань.

2.2 Дослідження та всесторонній аналіз методів вирішення проблеми масштабованості

Вирішення проблеми масштабованості блокчейн має вирішальне значення для широкого впровадження технології блокчейн. Різні проекти пропонують рішення, що мають за мету облегчення використання блокчейн мереж. Було проведено дослідження цих напрямків та проаналізовано їх сильні сторони та обмеження у застосуванні.

2.2.1 Масштабування другого рівня

Рішення для масштабування другого рівня – це інноваційні методи, які працюють поверх основного блокчейн, дозволяючи здійснювати транзакції поза мережею. Вони спрямовані на суттєве збільшення пропускної здатності транзакцій і зниження витрат на транзакції, зберігаючи безпеку та децентралізацію базового блокчейн.

Плазма (Ethereum). Плазма — це платформа для створення масштабованих автономних блокчейн смарт-контрактів, які часто називають «дочірніми ланцюжками», які періодично прив’язуються до основної мережі Ethereum [45]. Плазмові ланцюжки працюють незалежно, здійснюючи великий обсяг транзакцій. Періодично їхній стан прив’язується до основної мережі Ethereum, забезпечуючи безпеку та децентралізацію.

Ролапи (Rollups). Ролапи — це рішення другого рівня [46]. Вони обробляють транзакції, стискаючи дані, поза основним ланцюгом та надсилають остаточний стан в основний блокчейн. Дані рішення можуть значно збільшити пропускну здатність транзакцій і знизити плату за газ. Двома поширеними типами являються оптимістичні ролапи та zk-ролапи.

- Оптимістичні ролапи — це рішення для масштабування другого рівня, яке припускає, що більшість транзакцій є дійсними за замовчуванням, і надсилає дані лише в основний блокчейн (перший рівень) у разі суперечок або викликів. Такий підхід зменшує обсяг даних, які необхідно обробити в основному блокчейн. У оптимістичному ролапі транзакції

проводяться поза мережею в рамках смарт-контракту в мережі другого рівня. Користувачі можуть взаємодіяти зі смарт-контрактом другого рівня для швидких і недорогих транзакцій.

- zk-ролапи — це ще одне рішення для масштабування другого рівня, яке використовує передові криптографічні методи для забезпечення дійсності та цілісності транзакцій поза мережею. На відміну від оптимістичних зведень, zk-ролапи не покладаються на доступність даних транзакцій, зменшуючи ризик маніпулювання даними. zk-ролапи також обробляє транзакції поза мережею в рамках смарт-контракту другого рівня. Однак вони використовують криптографічні докази, щоб продемонструвати дійсність транзакцій, які потім передаються в основний блокчейн.

Стейт канали (State Channels). Стейт канали – це загальна концепція, застосовна до різних блокчейн-платформ [47]. Вони забезпечують взаємодію та транзакції поза ланцюгом.

Стейт канали можна класифікувати як рішення для масштабування другого рівня, якщо вони використовують основний блокчейн для остаточного розрахунку та безпеки. У цьому випадку учасники стейт каналу домовляються про стан каналу поза ланцюгом, але остаточний стан фіксується в блокчейн під час закриття каналу або у разі суперечок. Це гарантує безпеку та цілісність транзакцій каналу, підтверджену основною мережею.

Сайдчейни. Сайдчейни — це незалежні блокчейн, які сумісні з основним блокчейном, дозволяючи активам переміщатися між ними [48]. Користувачі можуть блокувати активи в основному блокчейн та передавати їх у бічний ланцюг для швидших і економічно ефективніших транзакцій. Сайдчейн підтримує свій реєстр та механізм консенсусу.

Аналіз переваг та недоліків наведено у додатку Б.

Підводячи підсумок, можна сказати, що рішення для масштабування другого рівня забезпечують цінний підхід до вирішення проблем масштабованості блокчейн, але вони також створюють складності та виклики,

пов’язані з досвідом користувачів, ліквідністю, централізацією, безпекою та сумісністю.

2.2.2 Рішення поза мережею

Рішення поза мережею передбачають проведення транзакцій і взаємодій повністю поза основним блокчейном. Ці транзакції відбуваються поза мережею, тобто вони не реєструються в реєстрі блокчейн.

Приклади рішень поза мережею включають платіжні канали (Lightning Network для Bitcoin) і стейт канали (Raiden Network для Ethereum). Вони забезпечують швидкі та недорогі транзакції між користувачами та можуть використовуватися для різних випадків використання, включаючи мікроплатежі.

Lightning Network (Bitcoin). Мережа Lightning — це добре відоме рішення другого рівня, призначене для покращення масштабованості Bitcoin [49]. Воно дозволяє здійснювати транзакції поза мережею через мережу платіжних каналів. Користувачі створюють платіжні канали, блокуючи певну кількість біткойн у блокчейн. Тоді вони можуть проводити практично необмежену кількість транзакцій у межах каналу без розрахунків у мережі. Фінальний результат цих транзакцій розраховується в блокчейн, коли канал закривається.

Raiden Network (Ethereum). Мережа Raiden — це еквівалент Ethereum Lightning Network [50]. Він дозволяє здійснювати однорангові платежі та перекази токенів поза мережею. Raiden Network створює платіжні канали для транзакцій поза мережею. Користувачі можуть надсилати токени один одному в межах цих каналів, а чистий результат записується в блокчейн Ethereum, коли канал закривається.

Стейт канали (State Channels). Стейт канали, у своїй основній формі можуть бути рішеннями поза мережею. Вони дозволяють двом або більше сторонам проводити кілька транзакцій поза мережею, не публікуючи кожну транзакцію в основному блокчейн. Це зменшує навантаження на основний блокчейн і покращує масштабованість. Транзакції в рамках стейт каналу розраховуються лише в блокчейн, коли канал відкривається або закривається.

Аналіз рішень поза мережею наведено у таблиці додатку В.

Підводячи підсумок, можна сказати, що рішення поза мережею пропонують ефективні способи покращити масштабованість, швидкість і економічну ефективність мереж блокчейн. Однак вони створюють нові проблеми безпеки, доступності даних, складності та сумісності, якими необхідно ретельно керувати для успішного розгортання та впровадження цих рішень.

2.2.3 Паралельний блокчейн

Паралельні блокчейн передбачають створення окремих блокчейн, з'єднаних з основним блокчейном. Кожен паралельний ланцюг може мати свій механізм консенсусу, правила та функції, зберігаючи зв'язок із основним ланцюгом для передачі активів і взаємодії.

Polkadot. Polkadot — це мережа з кількома ланцюжками, що забезпечує зв'язок та взаємодію кількох блокчейн [51]. Його створив Гевін Вуд, один із співзасновників Ethereum, і запустив Web3 Foundation. Polkadot розроблено для вирішення проблем масштабованості, безпеки та взаємозв'язку в просторі блокчейн. Основна мета Polkadot — надати можливість різним блокчейнам обмінюватися інформацією. Це досягається за допомогою мережі взаємопов'язаних блокчейн, які називаються парачейнами, які можуть працювати як один з одним, так і з основним релейним ланцюгом Polkadot.

Cosmos. Cosmos — це блокчейн-платформа, призначена для вирішення проблеми масштабованості в екосистемі блокчейн шляхом створення мережі взаємопов'язаних блокчейн [52]. Він спрямований на покращення масштабованості, сумісності та налаштування, зберігаючи безпеку та децентралізацію. Проект був ініційований Interchain Foundation, швейцарським некомерційним фондом, створеним для підтримки розробки та дослідження блокчейн-програмного забезпечення з відкритим кодом, однією з центральних фігур у її розробці був Дже Квон. Cosmos часто називають «Інтернетом блокчейн» через його фокус на з'єднанні кількох блокчейн. Він використовує

унікальну архітектуру, що складається з центрального концентратора, відомого як Cosmos Hub, і набору незалежних блокчейн, які називаються «зонами». Ці зони можна налаштувати за допомогою різних механізмів консенсусу та структур управління, що робить їх придатними для конкретних випадків використання.

Дослідження особливостей паралельного блокчейн наведено у таблиці додатку Г.

Таким чином, паралельні блокчейн пропонують ефективне вирішення проблеми масштабованості в мережах блокчейн шляхом розподілу транзакційного навантаження та сприяння налаштування та взаємодії. Однак вони створюють складності, пов’язані з безпекою, управлінням і сумісністю, якими необхідно ретельно керувати, щоб забезпечити успіх і стабільність усієї екосистеми блокчейн.

2.2.4 Шардинг

Шардинг — це важливий інноваційний підхід для вирішення проблем масштабованості в технології блокчейн [38]. Він передбачає поділ мережі блокчейн на менші частини, які називаються «шардами», для більш ефективної обробки транзакцій і смарт-контрактів. Опис переваг та недоліків даного підходу розглянуто у додатку Д.

Підсумовуючи, шардинг є перспективним рішенням для масштабованості блокчейн, що пропонує потенціал для збільшення пропускної здатності транзакцій, швидших підтверджень, нижчих комісій і покращеної децентралізації. Однак це також створює проблеми, пов’язані зі складністю мережі, безпекою, зв’язком між сегментами та доступністю даних.

2.2.5 Зміни консенсусного алгоритму

Деякі блокчейн переходять від енергоємних консенсусних алгоритмів, таких як Proof of Work, до більш ефективних і екологічно чистих алгоритмів, таких як Proof of Stake або Delegated Proof of Stake (DPoS).

Ethereum 2.0. Ethereum 2.0, також відомий як Eth2 або Serenity, є значним оновленням блокчейн Ethereum, яке спрямоване на перехід від механізму консенсусу PoW до механізму PoS [53]. Цей перехід обумовлений метою зменшення споживання енергії та підвищення масштабованості.

Ethereum 2.0 розгортається в кілька етапів, щоб забезпечити плавний перехід:

1) *Фаза 0 (Beacon Chain).* Фаза запущена в грудні 2020 року, представила Beacon Chain, ядро PoS Ethereum 2.0. Валідатори почали робити ставку на Beacon Chain для захисту мережі.
2) *Фаза 1 (Shard Chains).* Ця фаза реалізує ланцюжки фрагментів, що дозволяє паралельно обробляти транзакції. Очікується, що це покращить масштабованість мережі.
3) *Фаза 1.5 (Merge).* Це момент, коли Ethereum переходить від PoW до PoS. Основна мережа Ethereum стане фрагментом мережі Ethereum 2.0. Ця фаза ще знаходиться на стадії розробки.
4) *Фаза 2 (eWASM та середовища виконання).* Фаза принесе додаткові вдосконалення, включаючи інтеграцію eWASM і середовища виконання, що забезпечить більш гнучку та масштабовану розробку смарт-контрактів.

Tezos (DPoS). Tezos — це блокчейн-платформа, яка працює на основі механізму консенсусу DPoS [54]. DPoS є різновидом консенсусного алгоритму Proof of Stake і розроблений для вирішення проблем масштабованості та енергоефективності, які часто пов’язані з традиційними блокчейн Proof of Work.

Ефективність та слабкі сторони змі консенсусних алгоритмів можна переглянути у таблиці додатку Ж.

Таким чином, консенсусні зміни алгоритму є фундаментальним підходом до вирішення проблеми масштабованості в блокчейн-мережах. Вони пропонують потенціал для покращеної масштабованості, енергоефективності та безпеки, але вони також мають проблеми, такі як розгалуження мережі,

технічна складність і опір з боку спільноти. Ретельне планування та виконання є важливими для успішного переходу до нового механізму консенсусу.

2.2.6 Оптимізації

В ході роботи було вивчено перспективні напрямки дослідження та розробки, що зосереджені на оптимізації існуючих механізмів консенсусу, структур даних і криптографічних методів для підвищення ефективності блокчейн. Наведемо короткий огляд кожного підходу.

2.2.6.1 Segregated Witness

Segregated Witness (SegWit) [56], який часто називають абревіатурою SegWit, — це оновлення протоколу, призначене для вирішення проблеми масштабованості в блокчейн-мережах. Спочатку він був представлений як софтфорк у мережі біткойн, але згодом був прийнятий кількома іншими блокчейн-платформами. SegWit насамперед зосереджується на підвищенні ефективності транзакцій і зменшенні перевантаження мережі. Segregated Witness відокремлює дані транзакції від цифрових підписів (даних свідків) у транзакції блокчейн. У традиційній транзакції блокчейн дані транзакції та дані-свідки об’єднуються в єдину структуру даних. У SegWit ці два компоненти розділені, а дані-свідки зберігаються в окремій структурі даних. Детальніше ознайомитися з його особливостями можна у додатку К.

Segregated Witness відіграв вирішальну роль у покращенні масштабованості та ефективності мереж блокчейн, особливо у випадку з біткойн. Це стало стандартною функцією в біткойн і продовжує вивчатися та використовуватися іншими блокчейн-платформами як засіб вирішення проблеми масштабованості та покращення взаємодії з користувачем.

2.2.6.2 Докази з нульовим знанням

Докази з нульовим знанням (ZKP) [57] відіграють вирішальну роль у вирішенні проблем масштабованості в технології блокчейн. Дозволяючи

безпечні та приватні транзакції без розкриття конфіденційної інформації, ZKP сприяють розвитку мереж блокчейн.

ZKP використовують передові криптографічні методи для підтвердження автентичності транзакцій без розкриття конфіденційних даних, забезпечуючи безпечні та захищені від втручання операції. Впровадження ZKP полегшує перевірку транзакцій поза мережею, зменшуючи обчислювальне навантаження на основний блокчейн і покращуючи масштабованість. Опис переваг та недоліків наведено у додатку Л.

Підсумовуючи, хоча ZKP представляють деякі проблеми, їхні переваги з точки зору конфіденційності, масштабованості та безпеки роблять їх перспективним напрямком для покращення технології блокчейн. Триваючі дослідження та розробки в цій галузі продовжують удосконалювати та оптимізувати впровадження технології доказів з нульовим знанням для більш широкого впровадження.

ZKP вважається перспективним напрямком вирішення проблеми масштабованості блокчейн порівняно з іншими підходами завдяки кільком ключовим перевагам. Так, протоколи технології доказів з нульовим знанням пропонують унікальне поєднання масштабованості та конфіденційності. Вони забезпечують ефективну перевірку транзакцій, зберігаючи конфіденційність деталей транзакцій. Це має вирішальне значення для задоволення зростаючого попиту на конфіденційність транзакцій блокчейн без шкоди для продуктивності мережі [57]. ZKP також забезпечують належний рівень безпеки, адже вони покладаються на добре встановлені криптографічні принципи. Крім того, існують дуже ефективні протоколи з точки зору обчислювальних вимог. Вони дозволяють отримувати короткі та компактні докази, зменшуючи обчислювальне навантаження на мережу. Ця ефективність має важливе значення для підвищення швидкості обробки транзакцій і загальної масштабованості блокчейн. Також рішення ZKP не прив’язані до конкретного блокчейн або консенсусного алгоритму. Їх можна інтегрувати в різні блокчейн-платформи, що робить їх універсальним рішенням для

масштабованості. Ця сумісність гарантує, що протоколи ZKP можуть принести користь багатьом блокчейн-мережам.

Хоча ZKP пропонують безпечний інструмент для підвищення ефективності блокчейн мереж, важливо зазначити, що вони не позбавлені проблем і міркувань. Етап налаштування деяких протоколів та довіра до цілісності налаштування є важливими аспектами. Крім того, ZKP можуть вимагати коригування існуючої інфраструктури блокчейн, і їх впровадження може відрізнятися на різних платформах.

Отже, рішення доказів з нульовим знанням є багатообіцяючим підходом до вирішення проблем масштабованості блокчейн, пропонуючи збалансоване рішення, яке вирішує проблеми масштабованості та конфіденційності. Такі характеристики роблять даний підхід привабливим у дослідженні, тому саме його було обрано для вивчення та аналізу з точки зору перспективності та надійності застосування для подальшого впровадження в існуючі мережі блокчейн.

3 АНАЛІЗ ТА ПОРІВНЯЛЬНІ ДОСЛІДЖЕННЯ ВІДОМИХ ПРОТОКОЛІВ ДОКАЗІВ З НУЛЬОВИМ ЗНАННЯМ

Розвиток децентралізованих мереж став трансформаційною силою у світі фінансів та обміну інформацією. Ці мережі породили нові відносини, коли користувачі не підпорядковуються одному централізованому органу та можуть зберігати анонімність своїх дій. Значний внесок у цю еволюцію вносять криптографічні докази, які відіграють ключову роль у роботі та безпеці технології блокчейн, забезпечуючи цілісність транзакцій, перевірку автентичності даних, збереження конфіденційності та забезпечення децентралізованого консенсусу.

3.1 Дослідження концепцій криптографічних доказів

Криптографічні докази — це математичні чи обчислювальні методи, які надають докази чи гарантії правдивості чи дійсності твердження, твердження чи обчислення без розкриття основних деталей. Вони забезпечують безпечну взаємодію в ситуаціях, коли довіра обмежена або відсутня.

Дані алгоритми стали фундаментальними методами в галузі криптографії, які дозволяють перевіряти та запевняти конкретні твердження, не розкриваючи конфіденційну інформацію. Ці докази відіграють вирішальну роль у захисті цифрового зв’язку, перевірці особи та підтримці цілісності даних.

Криптографічні докази спрямовані на досягнення наступних властивостей у системі:

- Конфіденційність. Криптографічні докази дозволяють одній стороні довести знання секрету, не розкриваючи сам секрет. Це особливо важливо в програмах, орієнтованих на конфіденційність, як-от безпечний обмін повідомленнями та перевірка особи.
- Цілісність даних. Криптографічні докази дозволяють сторонам переконатися, що дані не були змінені чи підроблені під час передачі чи

зберігання. Це важливо для безпечної передачі та зберігання даних, особливо у фінансових транзакціях і технології блокчейн.

- Аутентифікація. Криптографічні докази використовуються для автентифікації, коли одна сторона підтверджує свою особу або авторизацію іншій стороні. Типовими прикладами є цифрові підписи та протоколи автентифікації. Вони гарантують, що відправник або одержувач повідомлення є тим, за кого себе видають.
- Безпека. Криптографічні докази забезпечують рівень безпеки, який робить обчислювально неможливим для зловмисників порушити валідність доказу або розкрити конфіденційну інформацію.

Криптографічні докази часто спираються на принципи теорії складності обчислень. Вони гарантують, що певні проблеми важко вирішити навіть для добре оснащених супротивників, що робить їх придатними для програм безпеки. Крім того, дані алгоритми розроблені таким чином, щоб бути ефективними, з низькими витратами на обчислення, що має вирішальне значення для реальних додатків.

Криптографічні докази бувають різних форм і служать різним цілям у сфері безпечного зв’язку та цілісності даних. Роздивимося кілька поширених варіантів криптографічних доказів [58]:

1) Докази з нульовим знанням – Zero-knowledge proofs (ZKP). Ці докази дозволяють одній стороні довести іншій, що вони знають секрет, не розкриваючи його сутності. ZKP використовуються для підтвердження особи, транзакцій із збереженням конфіденційності тощо.
2) Доказ знань – Proof of Knowledge (PoK). Більш загальна категорія, яка включає ZKP. PoKs демонструють не лише валідність твердження, але й те, що особа, яка перевіряє, має знання про те, як створити доказ. Ця концепція є основоположною в різних криптографічних протоколах.
3) Статистичні докази. Ці криптографічні докази допускають невелику ймовірність помилки. Вони забезпечують високий рівень впевненості, але

допускають можливість випадкових неточностей. Статистичні докази використовуються, коли повна впевненість не потрібна.

4) Квантові докази. Із розвитком квантових обчислень стали важливими квантово-стійкі криптографічні докази. Вони розроблені, щоб протистояти атакам квантових комп’ютерів, які можуть зламати багато класичних криптографічних схем.
5) Гомоморфні докази. Вони використовуються в гомоморфному шифруванні, техніці, яка дозволяє виконувати обчислення із зашифрованими даними без їх попереднього розшифрування. Гомоморфні докази забезпечують цілісність обчислень, зберігаючи конфіденційність даних.
6) Кільцеві підписи. Кільцеві підписи — це тип криптографічного доказу, який використовується в цифрових підписах. Вони дозволяють підписувачу створити підпис від імені групи, не розкриваючи, хто з учасників групи створив підпис. Це часто використовується в криптовалютах, орієнтованих на конфіденційність.
7) Багатостороннє обчислення (MPC). Хоча протоколи MPC не є доказом у традиційному розумінні, вони дозволяють кільком сторонам спільно обчислювати функцію на своїх вхідних даних без розкриття вхідних даних один одному. Вони відіграють вирішальну роль у безпечному обміні даними та спільному обчисленні.

Наведені варіанти криптографічних доказів є важливими інструментами сучасної криптографії та безпечного зв’язку. Вони забезпечують безпечні транзакції, конфіденційність даних, перевірку особи та цілісність даних у різних програмах, починаючи від криптовалют і блокчейн та закінчуючи безпечним обміним повідомленнями та аналізом даних.

Щоб зрозуміти роль криптографічних доказів у підвищенні масштабованості систем блокчейн, важливо переглянути фундаментальну роботу цих мереж. У міру розширення екосистем блокчейн виникає проблема

масштабованості, оскільки традиційний підхід, у якому кожен вузол перевіряє кожну транзакцію, стає неефективним.

Різноманітні ініціативи, включаючи такі проекти, як Aztec [60] і Starknet [61], вирішують цю проблему масштабованості шляхом використання криптографічних доказів. Ці рішення підпадають під категорію ролапів другого рівня (Layer 2 Validity Rollups), оскільки вони працюють на базовому блокчейн першого рівня, наприклад Ethereum. Фундаментальна концепція цих ролапів полягає в тому, що вони покладаються на криптографічні докази для підтвердження точності обчислень, проведених поза мережею, а також дійсності оновлень стану та наборів транзакцій [59].

Ця зміна парадигми є революційною для розподілених мереж, оскільки усуває необхідність сліпо виконувати кожен обчислювальний крок. Натомість мережа, що складається з багатьох стандартних машин, може просто перевірити ці кроки. Це означає, що більше суперкомп'ютери не потрібні – навіть звичайних, менш потужних комп'ютерів буде достатньо.

Вирішальне завдання перевірки цілісності масштабних обчислень покладено на надійні вузли мережі першого рівня. Отже, валідатори в мережі першого рівня беруть на себе роль «єдиного надійного ПК», а доказуючий другого рівня діє як свого роду суперкомп'ютер.

Таким чином, криптографічні докази дозволяють мережам блокчейн обробляти більший обсяг транзакцій та обчислень, зберігаючи при цьому безпеку, довіру та децентралізацію. Розвантажуючи та ефективно перевіряючи операції поза мережею, криптографічні докази сприяють постійному зростанню та масштабованості систем блокчейн.

3.2 Аналітичний опис напрямків доказів з нульовим знанням

Доказ з нульовим знанням – Zero-knowledge proof (ZKP) – це криптографічна концепція та протокол, що дозволяє одній стороні, яка називається доказуючим, довести іншій стороні, відомої як перевіряючий, що вони знають певну частину інформації або те, що твердження правдиве, без

розкриття самої фактичної інформації. Основна ідея доказів з нульовим знанням полягає в тому, що той, хто доводить, може переконати перевіряючого в істинності твердження, не розкриваючи жодної додаткової інформації про нього.

3.3.1 Дослідження загальних концепцій доказів з нульовим знанням

Протоколи з нульовим знанням покладаються на алгоритми, які приймають певні дані як вхідні дані та забезпечують вихід у двійковому вигляді «true» або «false» [62].

ZKP працює з трьома основними сутностями: твердження, доказуючий та перевіряючий. Твердження (Statement). Доказ нульового знання використовується для демонстрації дійсності твердження або факту без розкриття особливостей цього твердження. Наприклад, твердження може виступати паролем, криптографічним ключем або частиною даних. Доказуючий (Prover). Сторона, яка прагне довести достовірність твердження. Прувер має на меті переконати верифікатора в істинності твердження. Перевіряючий (Verifier). Сторона, яка отримує доказ, відома як верифікатор. Він перевіряє докази, надані доказувачем, щоб визначити, чи справді твердження є істинним.

Щоб вважатися дійсним протоколом з нульовим знанням, він повинен відповідати трьом основним критеріям:

- Повнота. Якщо вхідні дані протоколу валідні, а твердження дійсне, воно має постійно давати «істинний» результат. Тому, коли обидва твердження є істинними і обидві сторони (доказуючий та верифікатор) діють чесно, доказ слід прийняти.
- Достовірність. Якщо вхідні дані є невалідні, з надзвичайно малою ймовірністю нечесний доказуючий матиме змогу змусити протокол повернути «істинний» результат.
- Нульове знання. Верифікатор не повинен отримати жодних відомостей про саме твердження, крім його дійсності чи хибності – верифікатор повинен мати «нульові знання» про зміст твердження.

У мережах блокчейн, таких як Bitcoin та Ethereum, ZKP використовуються для покращення конфіденційності транзакцій, масштабованості та безпеки. Вона підвищує цілісність систем, одночасно захищаючи ідентифікаційні дані користувачів і деталі транзакцій. Користувачі мають змогу підтверджувати правдивість заяв, транзакцій та ідентифікаційних даних без необхідності розкривати додаткову інформацію. ZKP також дають змогу спільно аналізувати дані без шкоди для цілісності базових даних у сфері безпечних багатосторонніх обчислень. Це особливо важливо для сценаріїв, коли декільком сторонам потрібно спільно обробляти конфіденційні дані. Це створює потужний бар'єр проти невиправданого втручання в особисте життя та забезпечує надійний захист, необхідний для боротьби з кіберзлочинами, пов’язаними з особистими даними, забезпечуючи впевненість користувачів у безпеці їхніх ідентифікаційних даних в Інтернеті.

Для ілюстрації розглянемо сценарій, коли вам потрібно довести претензію, наприклад ваше громадянство, третій стороні, наприклад постачальнику послуг. Традиційно вам потрібно було б надати підтверджуючі докази, наприклад національний паспорт або водійські права [63]. Однак цей підхід має проблеми з конфіденційністю. Надання подібної інформації стороннім службам означає, що конфіденційні дані користувача зберігаються в централізованих базах даних, що робить їх уразливими для потенційних порушень. Зважаючи на зростання занепокоєння щодо крадіжки особистих даних, зростає попит на методи, які захищають конфіденційність під час обміну інформацією.

Докази з нульовим знанням пропонують геніальне рішення цієї скрутної ситуації, уникаючи необхідності розкривати інформацію для підтвердження тверджень. Протокол з нульовим знанням приймає заяву (яку називають «свідком») як вхідні дані та генерує стислий доказ її дійсності. Цей доказ пропонує надійні гарантії того, що твердження правдиве, не розкриваючи деталей, використаних для його побудови.

Доказ з нульовим знанням зробив значний стрибок у наданні користувачам розширеного контролю над своїми даними та фінансовою взаємодією в децентралізованих мережах. Вона підсилює концепцію суверенітету даних, дозволяючи людям відновити контроль над своєю особистою та фінансовою інформацією. Користувачі можуть диктувати, як, коли та з ким ділитися своїми даними, усуваючи залежність від централізованих організацій для керування інформацією.

Також технологія сприяє дотриманню нормативних вимог без шкоди для конфіденційності, не ставлячи при цьому під загрозу безпеку. Це дозволяє користувачам дотримуватися принципу KYC та інших нормативних вимог, не розкриваючи непотрібних особистих даних. Цей баланс між відповідністю та безпекою є ключовим у розвитку децентралізованих мереж.

Підводячи підсумок, Zero-Knowledge Proofs є передовою в трансформації того, як перевіряються ідентифікаційні дані в децентралізованих мережах. Вони надають користувачам безпечні механізми, орієнтовані на збереження конфіденційності, та ефективні засоби адаптації до цифрових платформ, одночасно зменшуючи ризики, пов’язані з крадіжкою особистих даних і шахрайством. Оскільки впровадження децентралізованих мереж продовжує зростати, роль технології ZKP у збереженні цілісності та конфіденційності цифрових ідентифікаційних даних стає все більш важливою, забезпечуючи безперервний розвиток безпечної та орієнтованої на користувача цифрової екосистеми.

3.3.2 Аналітичний огляд напрямків доказів з нульовим знанням

Алгоритми ZKP можна розділити на два види: інтерактивні та неінтерактивні докази. Перші працюють таким чином, що доказуючий та верифікатор беруть участь у зворотній взаємодії, де обмінюються низкою повідомлень. Неінтерактивні докази не потребують кількох раундів взаємодії. Пристрій перевірки може згенерувати єдине повідомлення, яке може перевірити верифікатор.

Докази з нульовим знанням бувають різних типів, кожен з яких призначений для конкретних випадків використання та вимог. Нижче наведемо аналіз поширених систем підтвердження нульового знання.

3.2.2.1 Протоколи Шнора та Сігми

Протоколи Шнора та Сігми (Schnorr and Sigma) [64]. Це інтерактивні ZKP, які демонструють знання секретного показника в групі, не розкриваючи самого показника. Вони використовуються в багатьох криптографічних протоколах і є основним будівельним блоком у ZKP. Вони використовуються для різних цілей, включаючи цифрові підписи, автентифікацію та підтвердження з нульовим знанням.

До сильних сторін протоколів Шнора та Сігма можна віднести наступне:

- Безпека. Вони забезпечують сильний опір різним криптографічним атакам, включаючи квантові атаки.
- Ефективність. Вони ефективні з точки зору перевірки, що сприяє швидшій обробці транзакцій у блокчейн.
- Лінійність. Підписи Шнорра є лінійними, що дозволяє їх агрегувати. Це означає, що кілька підписів можна об’єднати в один підпис, зменшуючи простір і обчислювальні вимоги. Це особливо важливо для покращення масштабованості мереж блокчейн.

До недоліків даного підходу можна віднести інтерактивність та складність. Протоколи Sigma є інтерактивними, вимагаючи кількох раундів зв’язку між прувером і верифікатором. Це може бути обмеженням у певних сценаріях, таких як транзакції блокчейн, де неінтерактивність є кращою для ефективності. Крім того, розробка протоколів Sigma та доказів з нульовим знанням на їх основі може бути складною та вимагати більше обчислювальних ресурсів.

Підсумовуючи, протоколи Schnorr і Sigma мають свої сильні та слабкі сторони. Підписи Schnorr відомі своєю ефективністю та лінійністю, що робить їх придатними для блокчейн. З іншого боку, протоколи Sigma пропонують

універсальність і гнучкість, але деякі реалізації можуть бути інтерактивними, що в деяких випадках може бути проблемним питанням.

3.2.2.2 Докази перемішування та діапазону

Докази перемішування та діапазону (Shuffle and Range) [65] — це спеціалізовані докази з нульовим знанням, призначені для демонстрації конкретних властивостей наборів даних, без розкриття основних деталей.

Метою доказу перемішування є демонстрація, що набір даних було перетасовано певним чином, не розкриваючи фактичний порядок елементів. Ці докази дозволяють верифікаторам підтверджувати процес перетасування, не знаючи позицій окремих елементів.

Ключовою перевагою доказів перемішування та діапазону є збереження конфіденційності, адже вони дозволяють перевіряти конкретні властивості даних без розголошення фактичних даних. крім того, вони забезпечують безпечну обробку конфіденційних даних та мають широкий спектр застосування – від фінансових операцій до підтвердження особи.

Однак, реалізація такий доказів може потенційно вимагати значної потужності обробки та часу, а їх розмір залежить від конкретних криптографічних методів, які використовуються, що в середовищах з обмеженими ресурсами може створювати проблеми,.

Отже, хоча докази Shuffle і Range пропонують значні переваги з точки зору конфіденційності та безпеки, їх впровадження може зіткнутися з проблемами, пов’язаними з обчислювальною складністю, розміром перевірки та необхідністю ретельного аналізу дизайну. Компроміси між конфіденційністю та ефективністю є центральними для оцінки цих криптографічних методів у різних програмах.

3.2.2.3 Bulletproofs

Bulletproofs [66]. Це неінтерактивні докази з нульовим знанням, відомі своєю ефективністю. Вони дозволяють перевіряльнику продемонструвати, що

значення потрапляє в певний діапазон, не розкриваючи точного значення. Вони зазвичай використовуються в криптовалютах, орієнтованих на конфіденційність.

Bulletproofs+. Спираючись на Bulletproofs, Bulletproofs+ покращує ефективність і гнучкість підтвердження діапазону значень, що робить його цінним вибором для криптовалют, які зберігають конфіденційність.

Перевагою даного підходу можна розглядати компактність – розмір доказу відносно малий, що важливо для ефективної перевірки та зберігання. Вони набагато ефективніші з точки зору розміру порівняно з деякими іншими криптографічними доказами. Bulletproofs є також ефективними з точки зору перевірки. Перевірку Bulletproofs можна виконати з відносно низькими обчислювальними витратами, що є вирішальним для реальних додатків, таких як технологія блокчейн. Крім того, Bulletproofs підтримують ефективне агрегування. Кілька Bulletproofs можна об’єднати в один доказ, зменшуючи загальний простір і обчислювальні вимоги. Це важливо для масштабованості в блокчейн-мережах, оскільки дозволяє проводити пакетну перевірку транзакцій. Bulletproofs дозволяють користувачам доводити правильність транзакцій, не розкриваючи суми транзакцій чи іншу конфіденційну інформацію. Це особливо цінно в криптовалютах, орієнтованих на конфіденційність.

З іншої точки зору, Bulletproofs є інтерактивними, що означає, що вони вимагають взаємодії між доказуючим та перевіряючим. Це може бути обмеженням у сценаріях, де неінтерактивність є кращою для ефективності. Розробка та впровадження Bulletproofs може бути складнішим порівняно з іншими системами перевірки, а створення Bulletproofs може потребувати інтенсивних обчислень. Більше того, розмір початкових перевірок Bulletproofs більший, ніж у деяких інших систем перевірки. Це може зробити початкове створення доказів більш ресурсомістким.

Підводячи підсумок, Bulletproofs є потужним інструментом для надання ефективних і конфіденційних доказів, головними перевагами яких є компактність, ефективність і масштабованість. Однак вони є інтерактивними,

що може не підходити для всіх випадків використання, і їхні початкові докази можуть бути більшими, ніж деякі альтернативи.

3.2.2.4 ZK-SNARK

ZK-SNARK (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge – короткий неінтерактивний аргумент знань із нульовим знанням) [67-68]. Це неінтерактивні ZKP, які дозволяють ефективно перевіряти обчислення без розкриття деталей цих обчислень. Вони широко використовуються в блокчейн-мережах, таких як Zcash і Ethereum, для покращення конфіденційності та масштабованості.

ZK-SNARK мають низку переваг:

- Конфіденційність. ZK-SNARK дозволяють користувачам доводити правдивість твердження, не розкриваючи подробиць цього твердження. Ця потужна функція конфіденційності є цінною для додатків, де конфіденційність даних має першорядне значення, наприклад для операцій із криптовалютою та автентифікації.
- Стислість. ZK-SNARK генерують короткі докази невеликого розміру, що робить їх ефективними як для передачі, так і для перевірки. Це важливо для масштабованості блокчейн та інших середовищ з обмеженими ресурсами.
- Масштабованість. ZK-SNARK підтримують пакетну перевірку, що означає, що кілька доказів можна перевіряти одночасно. Це об’єднання доказів покращує масштабованість мереж блокчейн та інших додатків, зменшуючи обчислювальне навантаження.
- Ефективність перевірки. Перевірка ZK-SNARK ефективна з точки зору обчислень. Верифікатори можуть швидко підтвердити достовірність доказів, зменшуючи обчислювальне навантаження на мережу.
- Неінтерактивность. ZK-SNARK є неінтерактивними, що означає, що вони не вимагають кількох раундів зв’язку між прувером і верифікатором. Ця характеристика спрощує їх використання в різних сферах застосування.

До недоліків даного підходу віднесемо довірене налаштування – чимало конструкцій ZK-SNARK передбачають фазу надійного налаштування, де довірена сторона генерує початкові параметри. Ця фаза може викликати занепокоєння щодо довіри, оскільки довірена сторона налаштування може потенційно поставити під загрозу безпеку. Крім того, вона може зайняти чимало часу та ресурсів. Цей процес налаштування може вимагати значної обчислювальної потужності та обережного поводження.

Таким чином, ZK-SNARK пропонують надійну конфіденційність, стислі докази та вдосконалення масштабованості, але вони мають проблеми з надійним налаштуванням.

3.2.2.5 ZK-STARK

ZK-STARK (Zero-Knowledge Scalable Transparent Argument of Knowledge – масштабований прозорий аргумент знань з нульовим знанням) [69]. ZK-STARK є неінтерактивними та відомі своєю масштабованістю та прозорістю. Вони забезпечують ефективну перевірку складних обчислень, що робить їх придатними для таких додатків, як рішення для масштабування блокчейн та безпечні системи голосування.

Перевагами даного підходу являються:

- Прозорість. ZK-STARK розроблені як прозорі, тобто для них не потрібна довірена фаза налаштування. На відміну від деяких інших систем перевірки, ZK-STARK не покладаються на довірену сторону для генерації початкових параметрів, що зменшує потенційні проблеми довіри.
- Безпека. ZK-STARK пропонують високий рівень безпеки. Вони розроблені, щоб протистояти різним криптографічним атакам, у тому числі тим, які викликані квантовими обчисленнями, що робить їх надійним вибором для додатків, які потребують конфіденційності та безпеки.
- Стислість. ZK-STARK генерують короткі докази, які мають відносно невеликий розмір. Це важливо для ефективної перевірки та зберігання,

особливо в блокчейн-мережах та інших середовищах з обмеженими ресурсами.

- Масштабованість. ZK-STARK підтримують пакетну перевірку, що дозволяє одночасно перевіряти кілька доказів. Ця можливість пакетної перевірки покращує масштабованість блокчейн-мереж та інших програм.

Недоліками виступають складність перевірки, інтерактивність та час ініціалізації. Перевірка ZK-STARK може потребувати інтенсивних обчислень і може вимагати більше обчислювальних ресурсів порівняно з іншими типами доказів, особливо коли твердження, що підтверджуються, складні. Хоча ZK-STARK не вимагає довіреного налаштування, процес генерації початкових параметрів для ZK-STARK може зайняти багато часу та ресурсів. Крім того, ZK-STARK не є інтерактивними за своєю конструкцією. Ця характеристика спрощує їх використання в деяких програмах, але може бути непридатною для сценаріїв, які потребують інтерактивних доказів.

Підсумовуючи, ZK-STARK пропонують покращення прозорості, безпеки та масштабованості, але вони мають такі проблеми, як складність перевірки та вимоги до ресурсів.

Кожен з наведених прикладів підтвердження нульового знання має свої сильні сторони, що робить їх придатними для різних випадків використання в різних сферах.

ZK-SNARK, зокрема, є перспективним напрямком для вирішення проблеми масштабованості блокчейн завдяки унікальному поєднанню переваг. По-перше, ZK-SNARK генерують невеликі лаконічні докази. Це має вирішальне значення для масштабованості блокчейн, оскільки зменшує розмір доказів, які потрібно зберігати та передавати через мережу. Менші докази означають менші вимоги до пам’яті та пропускної здатності, що покращує ефективність блокчейн.

По-друге, ZK-SNARK підтримують пакетну перевірку, дозволяючи одночасно перевіряти кілька доказів. Ця можливість зменшує обчислювальне навантаження під час перевірки численних транзакцій, що покращує

масштабованість блокчейн, дозволяючи йому обробляти більшу кількість транзакцій без істотного збільшення вимог до ресурсів. Перевірка ZK-SNARK також ефективна з точки зору обчислень. Ця ефективність сприяє швидшій обробці транзакцій, покращуючи загальну продуктивність блокчейн.

Крім того, ZK-SNARK є неінтерактивними, тобто вони не вимагають кількох раундів зв'язку між перевіряючим і верифікаційним. Така неінтерактивність спрощує процес перевірки та скорочує час, необхідний для підтвердження транзакції [68].

Єдиним суперечливим місцем технології є довірене налаштування. Хоча ZK-SNARK забезпечують конфіденційність і безпеку, довірене налаштування може потенційно викликати проблеми, адже якщо його компрометація підірве безпеку блокчейн.

Підводячи підсумок, ZK-SNARK добре підходять для вирішення проблеми масштабованості блокчейн завдяки своїй здатності надавати невеликі, ефективні з точки зору обчислень докази, дозволяючи їх пакетну перевірку. Ці характеристики дозволяють мережам блокчейн обробляти більший обсяг транзакцій, не перевантажуючи систему. Незважаючи на те, що ZK-SNARK потребує довіреного налаштування, дана технологія може ефективно впроваджуватися із належними заходами безпеки. Потенційні переваги, які пропонує ZK-SNARK з точки зору масштабованості та ефективності, роблять її переконливим рішенням для блокчейн-мереж, які прагнуть впоратися зі збільшеним навантаженням транзакцій.

3.3 Дослідження та аналіз протоколів напрямку ZK-SNARK

ZK-SNARK є одними з найбільш відомих і широко використовуваних типів SNARK. Вони пропонують стислі, неінтерактивні докази, які дозволяють ефективно перевіряти твердження, зберігаючи конфіденційність.

Для більшого розуміння протоколів доказів з нульовим знанням роздивимося спочатку основні етапи алгоритмів криптографічних доказів.

3.3.1 Вивчення технічних концепцій криптографічних доказів

Заглибимося в три технічні концепції, які лежать в основі всіх криптографічних доказах: арифметизація, застосування низького ступеня та криптографічні припущення.

Арифметизація. У сфері криптографічних доказів арифметизація передбачає перетворення математичних ідей і операцій в арифметичні операції, що виконуються в межах кінцевих полів. По суті, це включає вираження будь-якого заданого твердження у вигляді алгебраїчного рівняння, як правило, у формі полінома.

Арифметизація відіграє важливу роль у покращенні масштабованості, збільшуючи ймовірність виявлення будь-якого порушення цілісності, коли задачі виражаються мовою алгебри та поліномів [70].

Арифметизація є фундаментальною концепцією в криптографії, особливо в доказах з нульовим знанням. У криптографічних схемах використовуються різні підходи до арифметизації, кожен із яких має свої переваги та недоліки. Наведемо деякі поширені підходи до арифметизації:

- *Цілочисельна кільцева арифметизація.* У цьому підході проблеми представлені за допомогою модульної арифметики над цілочисельними кільцями.
- *Арифметизація еліптичної кривої*. Проблеми представлені за допомогою еліптичних кривих, які пропонують компактне та ефективне представлення.
- *Арифметизація на основі решітки.* Криптографія на основі решітки представляє проблеми за допомогою математичних структур, які називаються решітками.
- *Арифметизація на основі коду.* Криптографія на основі коду спирається на алгебраїчну теорію кодування для арифметизації проблем. Даний підхід постквантово безпечний і використовує сильні коди, що виправляють помилки.

- *Багатовимірна поліноміальна арифметизація.* Універсальний підхід, що може представляти широкий спектр задач, з використанням багатовимірних поліномів.
- *Програми квадратичної арифметики (QAP).* Математична структура, яка використовується в різних криптографічних протоколах, включаючи докази з нульовим знанням. Вони перетворюють обчислення на алгебраїчні рівняння, які можна ефективно обробляти та перевіряти. Знайшла своє використання у ZK-SNARK та ZK-STARK.
- *Модель арифметичної схеми (Arithmetic Circuit Model).* У цьому підході арифметизації обчислення представлені як арифметичні схеми, а система перевірки розроблена для роботи з цими схемами. Цей підхід дозволяє ефективно перевіряти обчислення, зберігаючи при цьому докази лаконічними. Використовується у Groth16.
- *Аргумент внутрішнього результату або доведення внутрішнього результату (Inner Product Argument або Inner Product Proofs).* Цей підхід арифметизації призначений для ефективного доведення скалярного добутку двох векторів без розкриття самих векторів. Даний тип арифметизації використовується у Bulletproofs.

Вибір підходу до арифметизації залежить від конкретних вимог до криптографічної схеми, включаючи міркування щодо безпеки, ефективності та характеру проблеми, що вирішується. У межах кожного підходу розроблено різні схеми, що пропонують ряд варіантів для різних застосувань.

Застосування поліномів низького ступеня. Застосування поліномів низького ступеня — це процес гарантування того, що поліноми (алгебраїчні рівняння, створені під час арифметизації) мають ступінь, нижчий за вказане порогове значення. Ступінь полінома відповідає найвищому ступеню члена в цьому поліному.

Застосування поліномів низького ступеня є важливим для підтримки цілісності та безпеки системи, адже поліноми нижчого ступеня менш схильні до певних вразливостей й атак. Крім того, у системах криптографічного доказу

ймовірність помилкового прийняття хибного доказу пов’язана зі ступенем вибраних поліномів, а використання поліномів низького ступеня знижує ризик прийняття хибних доказів [59].

Поліноми низького ступеня також забезпечують ефективність обчислень, зокрема прискореної верифікації. Це особливо важливо в додатках, де швидкість і ефективність використання ресурсів є критичними, наприклад транзакції блокчейн.

Застосування низького ступеня полінома є фундаментальним аспектом розробки безпечних і ефективних систем криптографічного підтвердження. Це допомагає знайти баланс між безпекою та ефективністю, гарантуючи, що докази є надійними та доступними для обчислень. Різні системи криптографічного підтвердження можуть мати різні вимоги та обмеження щодо низького ступеня полінома залежно від їх конкретних випадків використання та цілей проектування.

Схема поліноміального зобов’язання (polynomial commitment scheme, PCS). PCS — це криптографічний протокол, призначений для ефективного обчислення поліномів. У цій схемі доказуючий – коммітер (commiter), одна із залучених сторін – має можливість закріпити поліном, не розкриваючи його повних деталей. Згодом верифікатор, інша сторона, має можливість підтвердити властивості закріпленого полінома, не отримуючи доступу до його повної інформації.

Різні системи доказів використовують різні PCS для створення та перевірки доказів, найвідомішими є FRI та KZG.

FRI (Fast Reed-Solomon Interactive Oracle Proof) – це криптографічний протокол, розроблений для ефективної та короткої фіксації та перевірки великих поліномів. FRI було представлено як частину конструкцій ZK-SNARKs, особливо у контексті схем поліноміальних зобов'язань, які у ZK-SNARKs [71].

На етапі комітменту прувер генерує зобов'язання полінома високого ступеня, використовуючи рекурсивний процес, в якому вихідний поліном розбивається на компоненти нижчого ступеня. Обчислюється зобов'язання

кожному компоненту нижчого рівня, і процес повторюється рекурсивно до базового елемента.

Верифікатор може ефективно перевірити зобов'язання, використовуючи інформацію, одержану на етапі прийняття зобов'язання.

FRI досягає стислості за рахунок використання рекурсивної композиції розширень полінома низького ступеня, в результаті чого зобов'язання виходить набагато менше за розміром ніж зобов'язання вихідного полінома високого ступеня. Таке зменшення розміру має вирішальне значення для ефективності ZK-SNARK, де стисліть є ключовою вимогою.

Схема поліноміального зобов'язання KZG (названа на честь його початкових винахідників Кейт, Заверухи та Голдфедера) – це криптографічний протокол, який дозволяє ефективно виконувати фіксацію поліномів [72].

KZG дозволяє пруверу фіксувати поліном, використовуючи гомоморфні властивості, що дозволяє ефективно обчислювати зафіксовані поліноми, не розкриваючи їх.

Верифікатор може ефективно перевірити правильність зобов'язання, використовуючи короткий доказ, наданий доказуючим. Ефективність перевірки має вирішальне значення загальної ефективності системи доказів з нульовим розголошенням.

Криптографічні припущення. Криптографічні припущення – це математичні припущення або гіпотези, які складають основу безпеки криптографічних примітивів. Ці припущення включають складність певних математичних проблем [59].

Докази з нульовим знанням покладаються на криптографічні припущення для забезпечення безпеки та надійності системи доказів. ZK-SNARK припускають складність певних проблем: знанні покажчика ступеню (Groth16), модель алгебраїчної групи (PLONK, MARLIN), криптографії еліптичної кривої (Bulletproofs, Halo), стійкість до колізій хешів (STARK, Aurora тощо). Якщо ці проблеми обчислювально важко вирішити, ZKP залишається безпечним.

У контексті доказів із нульовим знанням наведені концепції працюють у гармонії. Арифметизація перетворює проблеми реального світу в математичні представлення, а застосування поліномів низького ступеня гарантує, що ці представлення піддаються обчисленням. Криптографічні припущення забезпечують базову структуру безпеки для доказів із нульовим знанням. Разом ці концепції дозволяють створювати ефективні, безпечні та конфіденційні системи доказів, які є цінними в широкому діапазоні програм.

3.3.2 Дослідження та аналіз протоколів ZK-SNARK

Нині сфера ZK-SNARK представляє різноманітні протоколи: GGPR (SNARK з попередньою обробкою) та Groth16, відомі своїм основоположним внеском у цій галузі, на ряду з пізнішими роботами: MARLIN, Aurora, PLONK тощо – зосереджуються на досягненні високої пропускної здатності та націлені на масштабованість у децентралізованих мережах, пропонуючи обслуговування різноманітних програм у різних варіантах використання.

Початкова система перевірки, яка заслуговує на увагу, — це стислий неінтерактивний аргумент знань – ZK-SNARK, який має компактний постійний розмір перевірки та витрати на перевірку, які залишаються незмінними навіть для значно великих зв’язків. Іншою важливою віхою є Groth16, ZK-SNARK, який відрізняється тим, що доказ містить лише три елементи групи. Примітно, що Groth16 зберігає високу ефективність у розмірі, порівнюючи з наступними досягненнями в цій галузі [73].

Однак, протокол вимагає надійне налаштування. Під час цієї фази довірена сторона генерує початкові параметри, які можуть викликати проблеми довіри. Якщо довірене налаштування зламано, це потенційно може поставити під загрозу безпеку системи. Іншим обмеженням Groth16 є те, що він функціонує як система перевірки із загальним посилальним рядком (CRS – common reference string) для конкретної схеми. Термін *загальний посилальний рядок* походить від моделі, що називається моделлю загального посилального рядка – моделі CRS, у якій і доказуючий, і верифікатор мають доступ до одного

рядка CRS, взятого з деякого існуючого дистрибутива. Загальний посилальний рядок зазвичай генерується на етапі довіреного налаштування та зачастішу використовується в неінтерактивних доказах з нульовим знанням [74]. Використання його означає, що система доказу адаптована для підтримки конкретної схеми на етапі налаштування пристрою для перевірки. Отже, якщо система доказу застосовується в різних програмах з різними схемами, це вимагає повторного запуску етапу налаштування з використанням інших параметрів.

Намагаючись вирішити дану проблему, Грот та інші науковці представили концепцію універсального та оновлюваного структурованого посилального рядка, що описано в протоколі GKM+. *Універсальний загальний посилальний рядок* означає, що він дозволяє одній установці підтримувати всі схеми певного обмеженого розміру. Універсальний загальний посилальний рядок (universal CRS) також часто називають універсальним структурним посилальним рядком (universal SRS). *Оновлюваний загальний посилальний рядок* дозволяє відкритому та динамічному набору учасників додавати до нього таємну випадковість на невизначений термін [73]. Це нововведення дозволяє створювати єдину конфігурацію для розміщення всіх схем у межах певного розміру. Крім того, оновлюваність SRS дозволяє динамічній групі учасників нескінченно вносити таємні випадкові дії. Хоча це налаштування зберігає елемент довіри, воно підвищує впевненість у безпеці параметрів, вимагаючи лише, щоб попередній учасник анулював свою секретну випадковість, щоб SRS залишався безпечним. Так, універсальність загального посилального рядка дозволяє застосовувати однакові параметри в різних програмах, що використовують цей ZK-SNARK. А можливість оновлення загального посилального рядка дозволяє користувачам змінювати параметри в будь-який час, навіть після розгортання системи. Ця характеристика дає змогу недовірливим користувачам самостійно оновлювати параметри, зміцнюючи особисту впевненість у надійності системи.

Незважаючи на наявність доказів постійного розміру та часу перевірки, протокол потребує SRS, який зростає квадратично щодо елементів множення в

арифметичних схемах. Крім того, оновлення SRS включають квадратичне піднесення до степеня в групі, а перевірка вимагає лінійного числа спарювань еліптичної кривої. Незважаючи на те, що для перевірки та верифікації потрібен лише рядок лінійного розміру, специфічний для схеми для фіксованого відношення (а не всієї SRS), вилучення його з SRS передбачає дорогий процес виключення Гауса.

Sonic, ще один протокол, розроблений головним чином для вирішення проблеми масштабування SRS, містить структурований посилальний рядок із лінійним розміром відносно підтримуваних схем. На відміну від цього, GKM+ масштабується квадратично. Sonic використовує менший набір глобальних параметрів, що забезпечує зручне зберігання, оновлення та перевірку на персональному ноутбуці. Протокол пропонує два режими роботи: «допоміжний» режим, простіший і ефективніший, але залежний від ненадійних третіх сторін для агрегування доказів, і «недопоміжний» режим, який уникає пакетування або агрегації, але має більшу вартість. У непідтримуваному режимі верифікатор Sonic для кожного доказу потребує знання оцінки розрідженого полінома для відомого полінома. Хоча обчислення цього полінома вимагає від верифікатора важких зусиль, воно залишається розумним для перевіряючого або помічника.

Наступним розвитком ZK-SNARK сфери була поява протоколу MARLIN, заснований на використанні спарювань еліптичних кривих. Головною задачей MARLIN, крім підвищення ефективності протоколів Groth16 і Sonic, є вирішення проблеми SRS. Він розроблений як універсальний і постійно оновлюваний, представляючи рішення проблем масштабованості та ефективності в системах з нульовим знанням. SRS ефективно масштабується, гарантуючи, що він залишається практичним навіть із збільшенням розміру підтримуваних схем. За допомогою універсальної та оновлюваної SRS MARLIN прагне підвищити ефективність zkSNARK, роблячи їх більш придатними для різних застосувань. Можливість постійного оновлення дозволяє адаптуватися до нових вимог без шкоди для безпеки. Цей підхід у MARLIN контрастує з

іншими протоколами zkSNARK, такими як Groth16 і Sonic, зосереджуючись на досягненні більш керованого та масштабованого SRS.

PLONK (Permutations over Lagrange-bases for Ecumenical Nointeractive arguments of Knowledge) [75] — це система доказів з нульовим знанням, яка зробила значний внесок у поле ZK-SNARK. PLONK використовує SRS та методи перестановки для підвищення ефективності обчислень для прувера, що спрощує його роботу. PLONK використовує універсальну та оновлювану SRS. Цей підхід забезпечує підвищену гнучкість та усуває необхідність у довіреному налаштуванні, вирішуючи головну проблему ZK-SNARK. Універсальний та оновлюваний SRS підвищує універсальність протоколу порівняно з Groth16, який спирається на фіксований та неоновлюваний SRS, а також з MARLIN, для якого може знадобитися фіксований SRS, обмежуючи його адаптованість. Більше того, PLONK усуває необхідність у довіреному налаштуванні, спрощуючи процес розгортання та підвищуючи рівень довіри [76].

У той час як MARLIN спирається на систему обмежень R1CS, PLONK є елегантною системою обмежень на основі перестановок, що пропонує переваги в певних випадках використання. З іншого боку, PLONK може мати більший розмір доказів у порівнянні з MARLIN або Groth16, але це часто компенсується підвищеною ефективністю та продуктивністю протоколу у певних сценаріях.

Крім того, існує покращена версія PLONK під назвою TurboPLONK, яка продемонструвала суттєві покращення порівняно з Groth16, а тести показали, що у певних сценаріях вона може бути у 2,5 рази швидше. TurboPLONK позиціонується як універсальний SNARK, що передбачає універсальність та застосовність у різних сценаріях та обчисленнях.

На нашу думку, даний протокол заслуговує на окрему увагу і вивчення, оскільки він є багатообіцяючим завдяки своїй підвищеній ефективності, надійності і застосовності в різних випадках використання. У дослідженні ми розглянули деталі протоколу PLONK та його вдосконаленої версії TurboPLONK. Крім того, була розроблена та реалізована схема перевірки валідності ланцюга блоків для аналізу та подальшого застосування в реальних системах блокчейн.

3.3.3 Вивчення технічних особливостей протоколів PLONK та TurboPLONK

Дослідження присвячено ZK-SNARK, а саме на багатообіцяючих протоколах, відомих як PLONK і TurboPLONK [77]. Ці криптографічні протоколи привернули увагу своїм потенціалом для вирішення проблеми масштабованості в системах блокчейн.

PLONK потребує довіреної процедури налаштування, він реалізує «універсальну та оновлювану» довірену установку. Це передбачає два важливі аспекти: (i) довірена настройка застосовується до всієї схеми, а не окремої довіреної установки для кожної програми, (ii) надійне налаштування дозволяє брати участь кільком сторонам, повний список учасників не обов’язково визначати заздалегідь. Така конструкція сприяє великій кількості учасників, забезпечуючи практичну безпеку.

Вирішальним елементом у PLONK, схожому на сценарій QAP [78], є процес, який перетворює формулювання задачі «надати такі значення, щоб задана програма, давала певний результат» у задачу «надати набір значень, що задовольняє набору математичних рівнянь». Це перетворення досягається шляхом представлення P як схеми логічних вентилів (гейтів), що включає операції додавання та множення. Згодом це представлення перетворюється на систему рівнянь, де змінні відповідають значенням уздовж усіх проводів, а кожен вентиль асоціюється з окремим рівнянням.

Щодо воріт і проводів, існують два типи обмежень: обмеження воріт (gate constraint) – рівняння між проводами, приєднаними до одного вентиля і обмеження копіювання (copy constraint) – твердження про рівність різних проводів будь-де в схемі.

Потрібно створити структуровану систему рівнянь, яка зрештою зведе до дуже невеликої кількості поліноміальних рівнянь, щоб представити обидва обмеження.

У PLONK налаштування для цих рівнянь є наступними (*L – left, R – right, O – output, M – multiplication, C – constant*) [79]:

$$(Q_{Li})a_i + (Q_{Ri})b_i + (Q_{Oi})c_i + (Q_{Mi})a_ib_i + Q_{Ci} = 0 \qquad (3.1)$$

Кожен Q є константою. Константи в кожному рівнянні, як і кількість рівнянь різні для кожної програми. Кожне значення – *a, b, c* – змінна, надана користувачем: a_i – лівий вхідний провід, b_i – правий вхідний провід, c_i – вихідний провід.

Таким чином, отримуємо задачу, коли доказуючий доводить, що він має набір значень x_{ai}, x_{bi} і x_{ci}, які задовольняють набору рівнянь однакової форми.

Наступним кроком є використання полінома як математичного інструменту для інкапсуляції цілого ряду значень в один об’єкт. Поліноми можуть бути представлені у «формі оцінки», для цього використовується інтерполяція Лагранжа.

Тепер розглянемо рівняння:

$$A_0(x)x_0 + A_1(x)x_1 + A_2(x)x_2 + \ldots + A_{n-1}(x)x_{n-1} - O(x) = Z(x)H(x) \qquad (3.2)$$

Кожен поліном Q є параметром, який може бути згенерований програмою перевіряючого, а поліноми $x_0, x_1, \ldots, x_{n-1}$ є вхідними даними користувача. $Z(x)$ — мінімальний (нетривіальний) поліном. Зауважте також, що в цьому випадку $H(x)$ зазвичай дорівнює нулю, але в більш складних випадках $H(x)$ може бути відмінним від нуля.

Тепер можна представити великий набір обмежень у невеликій кількості математичних об’єктів (поліномів). Змінні також можна представити у вигляді поліномів, щоб вони не були константними [79].

$$(Q_{Li})a(x) + (Q_{Ri})b(x) + (Q_{Oi})c(x) + (Q_{Mi})a(x)b(x) + Q_C(x) = 0 \qquad (3.3)$$

Кожен поліном Q є параметром, який може бути згенерований програмою перевіряючого, а поліноми *a, b, c* є вхідними даними користувача.

Всі обчислення виконуються над простим полем. Замість представлення індексів проводів за допомогою $x = 0,...n-1$, використовуємо w: $w^0, w^1,...,w^{n-1}$, де w – корінь одиниці в полі.

В ході протоколу необхідно перевірити задоволеності обмежень на вентилі – вираження (3.3).

Далі необхідно перевірити наступні обмеження на копіювання:

$$\begin{gathered} P_a(wx) - P_a(x)(v_1 + x + v_2 a(x)) = Z(x)H_1(x) \\ P_{a'}(wx) - P_{a'}(x)(v_1 + \sigma_a(x) + v_2 a(x)) = Z(x)H_2(x) \\ P_b(wx) - P_b(x)(v_1 + gx + v_2 b(x)) = Z(x)H_3(x) \\ P_{b'}(wx) - P_{b'}(x)(v_1 + \sigma_b(x) + v_2 b(x)) = Z(x)H_4(x) \\ P_c(wx) - P_c(x)(v_1 + g^2 x + v_2 c(x)) = Z(x)H_5(x) \\ P_{c'}(wx) - P_{c'}(x)(v_1 + \sigma_c(x) + v_2 c(x)) = Z(x)H_6(x) \end{gathered} \tag{3.4}$$

Перевірити початкові та кінцеві обмеження поліноміального накопичувача:

$$\begin{gathered} P_a(1) = P_b(1) = P_c(1) = P_{a'}(1) = P_{b'}(1) = P_{c'}(1) = 1 \\ P_a(w^n)P_b(w^n)P_c(w^n) = P_{a'}(w^n)P_{b'}(w^n)P_{c'}(w^n) \end{gathered} \tag{3.5}$$

Надані користувачем поліноми: (i) $a(x)$, $b(x)$, $c(x)$ – відповідають за проводи, (ii) $P_a(x)$, $P_b(x)$, $P_c(x)$, $P_{a'}(x)$, $P_{b'}(x)$, $P_{c'}(x)$ – обмеження поліноміального накопичувача (накопичувачі координат), (iii) $H(x)$ та $H_1(x),...,H_6(x)$ – частки.

Спеціальні для програми поліноми, які доказуючий та верифікатор повинні обчислити заздалегідь: Q_L, Q_R, Q_C, Q_M, Q_C, які представляють вентилі в схемі (Q_C кодує загальнодоступні вхідні дані, тому його може знадобитися обчислити або змінити під час виконання), «поліноми перестановки» $\sigma_a(x)$, $\sigma_b(x)$, $\sigma_c(x)$, які кодують обмеження копіювання між вентилями a, b, c.

Верифікатору потрібно лише зберігати зобов'язання щодо цих поліномів. Єдиний поліном $Z(x)$, що залишився у наведених вище рівняннях, призначений для обчислення нуля в усіх цих точках. На щастя, можна вибрати, щоб зробити цей поліном дуже простим для оцінки: вибрати w в степені n рівний одиниці.

Таким чином, задачу задоволення програми було перетворено на задачу задоволення кількох рівнянь за допомогою поліномів. У PLONK є оптимізації, які дозволяють видалити багато поліномів у наведених вище рівняннях, щоб полегшити обчислення, адже самі поліноми, як специфічні параметри програми, так і дані користувача, є великими.

Наступне питання полягає в тому, як обійти великі поліноми, щоб зробити доказ коротким.

3.3.4 Вивчення технічних особливостей протоколу FRI

Подальше вдосконалення базується на використанні стандартизованого криптографічного компонента – «поліноміальне зобов'язання». PLONK використовує «зобов'язання Kate» на основі надійної установки та поєднання еліптичних кривих. Однак, в дослідженні його було замінено на FRI, що перетворює PLONK на своєгороду STARK. Ця можливість адаптації робить схему теоретично сумісною з різними компромісами між розміром доказу та припущеннями безпеки.

Щоб підтвердити, що всі точки належать поліному одного ступеня, необхідне комплексне дослідження кожної точки. Неможливість перевірити хоча б одну точку створює ймовірність того, що вона може не узгоджуватися з поліномом, навіть якщо інші збігаються. Однак імовірнісний підхід дозволяє перевірити, що певний відсоток, наприклад 90%, точок вірогідно відповідає одному поліному.

Щоб визначити належність цих точок, D точок недостатньо. Але можна створити поліном ступеня $<D$, для визначення якого необхідно D точок. Для цього самого поліному $<D$ можна перевірити $D+1$ точок. Спочатку необхідно вибрати випадкову підмножину точок D і застосувати інтерполяцію Лагранжа, щоб реконструювати унікальний поліном ступеня $<D$, що проходить через усі вибрані точки. Згодом випадковим чином можна вибрати одну додаткову точку та перевирити її суміщення з тим самим поліномом [80].

Це тест на спорідненість, який підтверджує сценарій, коли більшість точок узгоджується з поліномом низького ступеня, але деякі відхиляються, і можливо, $D+1$ пропустила якісь точки. Тим не менш, можна зробити висновок, що якщо менше ніж 90% точок відповідають поліному ступеня $<D$, тест, швидше за все, провален. Інакше кажучи, якщо при $D+k$ запитів частка p точок розходиться із загальним поліномом, ймовірність успіху тесту становить $(1-p)^k$.

Розглянемо сценарій, коли D є помітно високим і ми прагнемо перевірити ступінь полінома за допомогою кількості запитів $<D$. Пряма перевірка неможлива – будь-які точки $k \leq D$ знаходяться принаймні в одному ступені $<D$ полінома. Однак можна досягти цього шляхом введення додаткових даних. Це і є мета сучасних протоколів, таких як FRI або IOPP (Interactive Oracle Proofs of Proximity) тощо.

Почнемо з відносно простого протоколу з досить поганими компромісами, але це все одно досягає мети складності сублінійної перевірки, тобто можна довести наближеність до полінома ступеня $<D$ з менш ніж D запитами (менше, ніж $O(D)$ обчислення для перевірки доказу) [79].

Припустимо, що є N точок $N = 1$ мільярд, і всі вони на поліномі $f(x)$ ступеня <1000000. Знаходимо поліном двох змінних $g(x, y)$, такий, що $g(x, x^{1000})=f(x)$. Це можна зробити наступним чином: для члена k-го ступеня у $f(x)$ розкладаємо його на $x^{(k\%1000)}y^{(floor(k/1000))}$.

На першому етапі доказу доказуючий робить зобов'язання – створює дерево Меркла – обчислення $g(x,y)$ по всьому квадрату $(1...N]x\{x^{1000}:1\leq x\leq N\}$ - тобто всі один мільярд координат для стовпців і всі один мільярд відповідних тисячних степенів для координат у рядків. Діагональ квадрата представляє значення $g(x, y)$, які мають форму (x, x^{1000}), і таким чином відповідають значенням $f(x)$.

Потім верифікатор випадковим чином вибирає, можливо, кілька десятків рядків і стовпців (використовуючи квадратний корінь Меркла як джерело псевдовипадковості для отримання неінтерактивного доказу), і для кожного обраного рядка або стовпця верифікатор запитує вибірку, наприклад, з 1010

точок у рядку та стовпці, переконуючись у кожному випадку, що одна з запитуваних точок знаходиться на діагоналі. Доказубчий повинен відповісти, надавши ці точки, а також гілки дерева Меркла, доводячи, що дані є частиною даних, наданих даказуючим при зобов'язанні. Верифікатор перевіряє, чи збігаються гілки Меркла, і що точки, які надає прувер, справді відповідають поліному ступеня *1000*.

Наведений алгоритм доводить дві речі:

- Більшість рядків та стовпців заповнено переважно точками поліномів ступеня <1000,
- Діагональна лінія здебільшого з наведених поліномів.

Таким чином, це переконує верифікатора, що більшість точок на діагоналі дійсно відповідають поліному ступеня <1000000. Потрібно в рази менше даних: якщо прувер робить зобов'язання на тридцять рядків та стовпці, а верифікатору *1010* точок, то загалом потрібно *60600* точок. Робота доказуючого займає чималих ресурсів, адже йому необхідно зафіксувати весь $N \cdot N$ прямокутник. Хоча поліноміальну інтерполяцію можна зробити субквадратичною, алгоритм у цілому є сублінійним для перевірки.

Тепер перейдемо до складнішої версії протоколу, яка має за мету зменшити складність перевірки з 10^{18} до 10^{15}, а потім до 10^9. По-перше, замість роботи зі звичайними числами будемо оперувати модульною математикою [80].

Зокрема, будемо працювати з простим модулем $p=1000005001$. Зведення до степеня x^{1000} матиме зображення розміром 1000006, тобто піднесення до степеня може дати лише 1000006 можливих результатів.

Це означає, що діагональ (x, x^{1000}) тепер стає діагоналлю з обертанням, оскільки x^{1000} може приймати лише 1000006 можливих значень, нам потрібно лише 1000006 рядків. Отже, повна оцінка $g(x, x^{1000})$ тепер містить лише 10^{15} елементів.

Тепер можемо змусити доказуючого виконувати оцінку g для одного стовпця. Ключовий трюк полягає в тому, що вихідні дані самі по собі вже містять *1000* точок у будь-якому даному рядку, тож можна просто взяти їх на

вибірку, отримати поліном ступеня <1000, на якому вони знаходяться, а потім перевірити, чи знаходиться відповідна точка у стовпці для певного поліному. Потім перевіряємо, що сам стовпець це поліном ступеня $a<1000$.

Складність доказуючого все ще є сублінійною, але складність верифікатора тепер зменшилася до 10^9, що робить його лінійним за кількістю запитів.

У цьому розділі було розглянуто математичні концепції, що лежать в основі протоколів PLONK і FRI, підкресливши їх ефективність як ключове рішення проблеми масштабованості блокчейна. Спираючись на цю основу, дослідження перейшло до фази впровадження, де було використано наведені протоколи для розробки надійної системи перевірки обчислювальної цілісності ланцюжка взаємопов'язаних блоків. Інтеграція протоколів PLONK і FRI в схему тестування мала за мету підвищення безпеки та надійності мереж блокчейн, гарантуючи, що обчислювальні процеси кожного блоку підтримують очікувані стандарти цілісності. Застосування передових математичних протоколів не тільки вирішує проблеми масштабованості, але й створює стійку та надійну основу для технології блокчейн, сприяючи її подальшому розвитку та прийняттю в різних областях.

4 РОЗРОБКА ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ ПЕРЕВІРКИ ВАЛІДНОСТІ ЛАНЦЮГА БЛОКІВ ІЗ ЗАСТОСУВАННЯМ ТЕХНОЛОГІЇ ZK-SNARK

Для вирішення проблеми масштабованості блокчейн та, зокрема позбавлення від нативної перевірки реєстру – перерахунку блоків мережі, в дослідженні використовується технологія ZK-SNARK, а саме протоколи PLONK і TurboPLONK у поєднанні зі схемою зобов’язань FRI.

ZK-SNARK прискорює генерацію доказів. Так, для обчислень, які потребують *t* кроків, можна зробити доказ приблизно за $O(t \cdot log(t))$. Перевірка такого доказу вимагає $O(log2(t))$ кроків.

У цьому розділі представлено розроблену схему для рекурсивного доказу обчислювальної цілісності ланцюжка пов’язаних блоків. У схемі використовується пов’язаний список, побудований через криптографічний зв’язок. Для кожного блоку ми формуємо докази обчислювальної цілісності хешів і цифрових підписів, що входять до блоку.

Реалізуючи схему доведення, було вирішено провести дослідження в двох напрямках.

Перший, називається агрегацією. Для цієї схеми створюється ланцюжок пов’язаних хешів. Після цього створюються докази для перевірки хешу та цифрового підпису, а потім вони об’єднуються в одне підтвердження. Ланцюжок доказів створюється шляхом агрегування доказів попереднього блоку з поточним. У результаті для кожного конкретного блоку можна перевірити доказ того, що хеш блоку та підписи були обчислені правильно, а ланцюжок доказів для попередніх блоків було перевірено правильно. Цей метод вимагає цифрових підписів для підтвердження обчислювальної цілісності блоку, інакше можливий зловмисник може підмінити деякі блоки.

Друга ідея, яка називається рекурсією, базується на рекурсивно зв'язаному ланцюжку. Тобто створюється блок генезису. Після цього

генерується єдиний доказ для хешу та підписів поточного блоку, не за допомогою їх агрегації, а створенням збільшеної схеми. Доказ наступного блоку формується шляхом перевірки попереднього блоку (тобто шляхом перевірки правильності хешу та підписів попереднього блоку) та обчислення доказу для поточного блоку. Ця схема дозволяє перевірити попередній хеш, щоб вирішити проблему підміни блоку. Крім того, схема не потребує цифрового підпису для забезпечення обчислювальної цілісності ланцюжка.

4.1 Розробка схеми ланцюга доказів обчислювальної цілісності ланцюжка пов’язаних хешів

У блокчейн кожен наступний блок формується за допомогою криптографічного з'єднання, організованого хешуванням, як показано на рисунку 1.1.

Дані блоку залежать від архітектури блокчейн, але зазвичай в кожному блоці є кілька транзакцій, унікальний номер, а також хеш попереднього блоку. Вміст блоку (включаючи хеш попереднього блоку) хешується, а отримане хеш-зображення включається в наступний блок. Обчислювально складно знайти прообраз (змінити вміст блоку) для відомого хешу (недоступний на сучасних комп’ютерах), який забезпечує надійне зв’язування блоків у ланцюжок. Іншими словами, обчислювально недосяжно змінити історичні записи таким чином, щоб не порушити цілісність ланцюжка — зміна навіть одного біта у вмісті блоку призведе до зміни хеш-образу на 50%, що буде виявлені під час перевірки цілісності.

Для тестового прикладу розглянемо спрощений варіант (рисунок 4.1), коли кожен блок містить лише два поля: хеш попереднього блоку h_{i-1} та унікальний номер блоку *Nonce*.

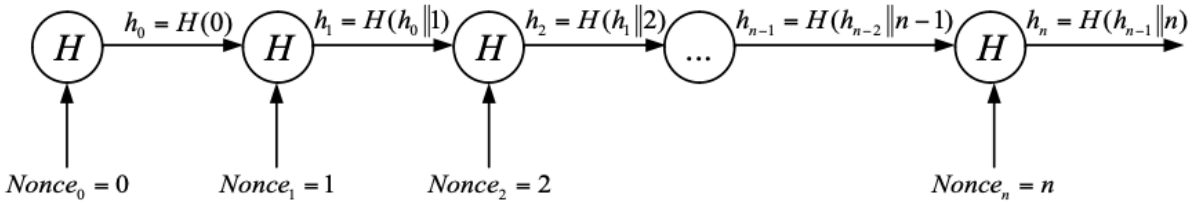

Рисунок 4.1 — Спрощена схема пов’язаного списку

Візьмемо $Nonce_i = i$. Потім для кожного i-го блоку виконуємо хешування $h_i = H(h_{i-1}||i)$ над результатом конкатенації значень h_{i-1} та i. Таким чином, результат n-го хешування можна представити наступним чином:

$$h_n = H(h_{n-1} \| n) = H(H(h_{n-2} \| n-1) \| n) = \ldots = H(H(H(\ldots H(H(0) \| 1) \ldots \| n-2) \| n-1) \| n) \qquad (4.1)$$

Завдання полягає в тому, щоб реалізувати тестовий приклад рекурсивного підтвердження обчислювальної цілісності (ОЦ) для ланцюжка пов’язаних хешів. Іншими словами, необхідно довести обчислювальну цілісність h_n за допомогою наведеного вище виразу.

Для цього необхідно послідовно реалізовувати наступні кроки:

1) Реалізація ланцюжка хешів $h_0 = H(0)$, $h_i = H(h_{i-1}||i)$, $i = 1,\ldots n$;
2) Створення схеми $C(x_i, w_i)$ для алгоритму хешування H для кожного $h_0,\ldots h_n$, $i = 1,\ldots n$, де $x_i = h_i$ — результат хешування, w_i — вхідні дані (свідок, хеш-прообраз) для обчислення хешу: $w_0 = 0$, $w_i = (h_{i-1}||i)$, $i = 1,\ldots n$;
3) Формування загальнодоступних параметрів для всіх $i = 0,\ldots n$: $(S_{pi}, S_{vi}) = S(C_i(x_i, w_i))$, де S_{pi} – публічні параметри перевірки, S_{vi} – загальнодоступні параметри перевірки;
4) Формування доказу ОЦ для всіх $i = 0,\ldots n$: $\pi_i = P(S_{pi}, x_i, w_i)$;
5) Реалізація алгоритму перевірки доказів $V(S_{vi}, x_i, \pi_i)$, результатом роботи якого є значення *{0, 1} (accept or reject)*;
6) Перевірка доказу, що $V(S_{vi}, x_i, \pi_i) = accept$ для всіх $i = 0,\ldots n$.

Результатом виконання завдань 1-6 є формування множини доказів π_i $(i = 0,\ldots n)$ обчислювальної цілісності, яка показана на рисунку додатку М.

Таким чином, сукупність наведених доказів доводить обчислювальну цілісність кожного хешу, а саме, що вхідні дані відповідають запропонованому хешу. Ця схема усуває можливість підміни хеша: якщо дані блоку було взято з довіреного джерела, та для них було сформовано доказ, а хеш запропоновано деяким користувачем, тоді можна перевірити за допомогою алгоритму верифікації $V(S_{vi}, x_i, \pi_i)$ відповідність хеша сформованому доказу, тобто його валідність.

Для підтвердження цілісності всього ланцюга необхідно рекурсивно з'єднати отримані докази в один для певного проміжку, що розглянуто у наступних підрозділах.

4.2 Розробка схеми валідності ланцюга блоків обчислювальної цілісності хешів: методи агрегації та рекурсії

Агрегація. Щоб створити ланцюжок пов'язаних хешів, тобто довести верифікатору V правильність обчислення $h_0 = H(0)$, $h_i = H(h_{i-1}||i)$, $i = 1,\ldots n$, розглянемо значення $\pi_i = P(S_{pi}, x_i, w_i)$ $(i = 0,\ldots n)$ як вхідні дані W_i під час агрегування доказів.

1) Формування обчислювальної схеми $C_i(X_i, W_i)$ алгоритму верифікації V для кожної пари доказів $\pi_{i-1} = P(S_{pi-1}, x_{i-1}, w_{i-1})$ та $\pi_i = P(S_{pi}, x_i, w_i)$, де $X_i = (V(S_{vi-1}, x_{i-1}, \pi_{i-1}), V(S_{vi}, x_i, \pi_i))$ – результати перевірки (accept or reject) доказів π_{i-1} та π_i. $W_i = (\pi_{i-1}, h_{i-1}, \pi_i, h_i)$ – вхідні дані для перевірки доказів;
2) Створення загальнодоступних параметрів $(S_{Pi}, S_{Vi}) = S(C_i(X_i, W_i))$, $(i = 1,\ldots n)$, де S_{Pi} – загальнодоступні налаштування верифікатора, S_{Vi} – загальнодоступні параметри перевірки;
3) Генерація доказу обчислювальної цілісності $\prod_i = P(S_{Pi}, X_i, W_i)$ для всіх $i = 1,\ldots n$;
4) Реалізація алгоритму перевірки доказів $V(S_{Vi}, X_i, \prod_i)$, який приймає значення *{0, 1} (accept or reject)*;
5) Реалізація алгоритму перевірки доказів, що $V(S_{Vi}, X_i, \prod_i) = accept$.

Таким чином, кожен доказ $\prod_i = P(S_{Pi}, X_i, W_i)$ є сукупністю двох інших доказів $\pi_{i-1} = P(S_{pi-1}, x_{i-1}, w_{i-1})$ та $\pi_i = P(S_{pi}, x_i, w_i)$, які надаються як вхідні дані (свідки) доказу $\prod_i$. Обчислювальна схема $C_i(X_i, W_i)$ побудована таким чином, що результат перевірки $V(S_{Vi}, X_i, \prod_i) = V(S_{vi-1}, x_{i-1}, \pi_{i-1})$ приймає значення $V(S_{vi}, x_i, \pi_i)$.

Якщо результати X_i перевірки доказів $V(S_{vi-1}, x_{i-1}, \pi_{i-1}) = accept$ та $V(S_{vi}, x_i, \pi_i) = accept$ безпосередньо використовуються при формуванні обчислювальної схеми, то лише свідок W_i можна подати на вхід схеми, тобто схема виглядає наступним чином: $C_i(W_i)$ без відкритих входів X_i.

Схема агрегації представлеа на рисунку додатку Н.

На рисунку додатку П показана інша версія схеми доказу, в якій:

1) $X_i = (V(S_{vi-1}, x_{i-1}, \pi_{i-1}), V(S_{vi}, x_i, \pi_i))$ для всіxl $i = 1,...n$.
2) $W_1 = (\pi_0, h_0, \pi_1, h_1)$, $W_i = (\prod_{i-1}, X_{i-1}, \pi_i, h_i)$ для всіх $i = 2,...n$.

У цьому випадку виконання умови $V(S_{Vi}, X_i, \prod_i) = accept$ для всіх $i = 2,...n$ означає, що перевірка доказу $V(S_{Vi-1}, X_{i-1}, \prod_{i-1})$ та $V(S_{vi}, x_i, \pi_i)$ були обчислені правильно. Якщо одночасно $V(S_{Vi-1}, X_{i-1}, \prod_{i-1}) = accept$ та $V(S_{vi}, x_i, \pi_i) = accept$, це означає, що:

1) Перевірка доказу $V(S_{Vi-2}, X_{i-2}, \prod_{i-2})$ і $V(S_{vi}, x_i, \pi_i)$ обчислена правильно;
2) Значення $h_i = H(h_{i-1}||i)$ обчислено правильно.

Таким чином, приклад другого ланцюга рекурсивних доказів агрегує в кожному $\prod_i$ всі попередні докази ($i = 0,...n$). Наприклад, останній доказ у ланцюжку $\prod_n = P(S_{Pn}, X_n, W_n)$ перевіряє правильність значення $h_n = H(h_{n-1}||n)$ і всіх попередніх доказів.

Для формування доказу $\prod_n = P(S_{Pn}, X_n, W_n)$ як вхідні дані використовуються:

1) $Xn = (V(S_{vn-1}, x_{n-1}, \pi_{n-1}), V(S_{vn}, x_n, \pi_n))$ – результати перевірки двох попередніх доказів;
2) $Wn = (\prod_{n-1}, X_{n-1}, \pi_n, h_n)$, тобто доказ $\prod_{n-1}$ і результат його перевірки та доказ π_n.

Умови $V = accept$ слід використовувати при формуванні всіх обчислювальних схем рекурсивних доказів. Тоді результат перевірки $V(S_{Vi}, X_i, \prod_i) = accept$ означає, що всі хеші в ланцюжку (рисунок 1.1) обчислені правильно, і всі перевірки для всіх $i = 0,...n$ також обчислені правильно.

Рекурсія. Щоб створити ланцюжок пов'язаних хешів $h_0 = H(0)$, $h_i = H(h_{i-1}||i)$, $i = 1,...n$, розглянемо значення $\pi_i = P(S_{pi}, x_i, w_i)$ ($i = 0,...n$) як вхідні дані W_i під час створення доказу для наступного блоку, схематично представлено на рисунку додатку Р:

1) Формування обчислювальної схеми $C_i(X_i, W_i)$, яка включає алгоритм верифікації V для попереднього блоку $\pi_{i-1} = P(S_{pi-1}, x_{i-1}, w_{i-1})$ та алгоритм хешування H для поточного блоку, де $X_i = (V(S_{vi-1}, x_{i-1}, \pi_{i-1}), x_i)$: $V(S_{vi-1}, x_{i-1}, \pi_{i-1})$ є результатом перевірки (accept or reject) попереднього доказу π_{i-1} та $x_i = h_i$ є результатом хешування. $W_i = (\pi_{i-1}, h_{i-1}, w_i)$: π_{i-1}, h_{i-1} – вхідні дані для перевірки доказу, а w_i – вхідні дані (свідок, хеш-прообраз) для обчислення хешів: $w_0 = 0$, $w_i = (h_{i-1}||i)$, $i = 1,...n$;
2) Створення загальнодоступних параметрів $(S_{Pi}, S_{Vi}) = S(C_i(X_i, W_i))$ ($i = 1,...n$), де S_{Pi} – загальнодоступні налаштування верифікатора, S_{Vi} – загальнодоступні параметри перевірки;
3) Генерація доказу обчислювальної цілісності $\pi_i = P(S_{Pi}, X_i, W_i)$ для всіх $i = 1,...n$;
4) Реалізація алгоритму перевірки доказів $V(S_{Vi}, X_i, \pi_i)$, що приймає значення $\{0, 1\}$ *(accept or reject)*;
5) Перевірка доказу, що $V(S_{Vi}, X_i, \pi_i) = accept$.

Таким чином, кожне доказ $\pi_i = P(S_{Pi}, X_i, W_i)$ формується на основі попереднього $\pi_{i-1} = P(S_{pi-1}, x_{i-1}, w_{i-1})$ і власного алгоритму хешування.

4.3 Розробка схеми валідності ланцюга блоків обчислювальної цілісності ланцюжка пов'язаних хешів із цифровим підписом

Представлений в даному розділі приклад реалізує рекурсивне підтвердження обчислювальної цілісності ланцюга зв'язаних хешів із

перевіркою цифрового підпису (ЦП). Це захищає від можливих змін даних і накладення фальшивого ланцюжка пов’язаних хешів.

Для тесту розглянемо спрощену версію з попередньої схеми, доповнену перевіркою цифрового підпису (рисунок 4.6). Кожен блок містить три поля:

- Унікальний номер блоку: $Nonce_i$;
- Хеш попереднього блоку: h_{i-1};
- Цифровий підпис: $EDS(h_i)$.

Для спрощення (як у попередньому випадку) приймаємо $Nonce_i = i$. Результат n-го хешування:

$$h_n = H(h_{n-1} \| n) = H(H(h_{n-2} \| n-1) \| n) = \ldots = H(H(H(\ldots H(H(0) \| 1) \ldots \| n-2) \| n-1) \| n) \qquad (4.2)$$

Додатково кожен хеш h_i шифрується секретним ключем sk, тобто ми формуємо підпис $EDS(h_i, sk)$. Відкритий ключ pk використовується для перевірки підпису, тобто ми розшифровуємо EDS_i та перевіряємо рівність $h_i = D(EDS_i, pk)$. Ми використовуємо лише алгоритм перевірки підпису $h_i = D(EDS_i, pk)$ для створення доказів і перевірки обчислювальної цілісності.

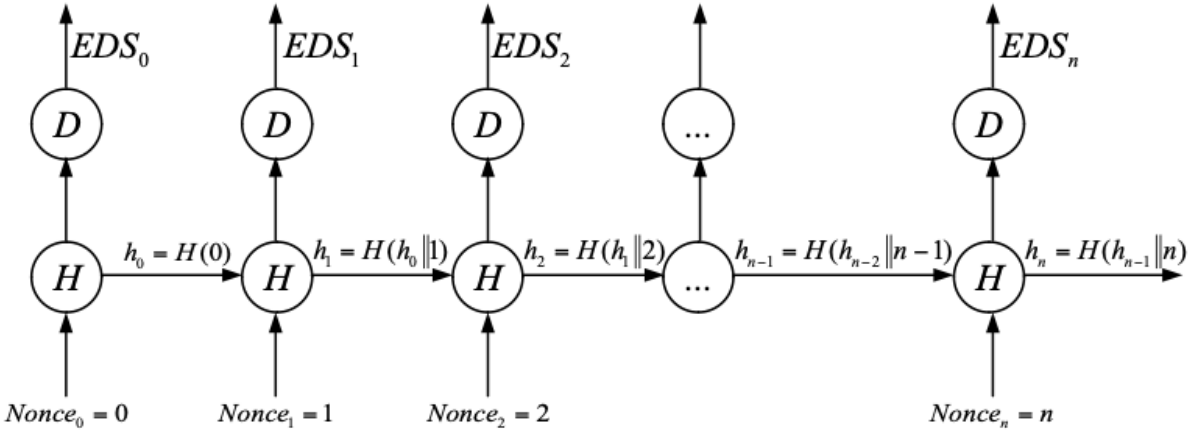

Рисунок 4.6 — Спрощений варіант зв’язаного списку з формуванням ЦП

Агрегація. Для реалізації рекурсивного доказу ОЦ ланцюга необхідно послідовно виконувати наступні етапи, відповідно до схеми на рисунку 4.7 додатку С:

1) Релаізація ланцюга хешів $h_0 = H(0)$, $h_i = H(h_{i-1}||i)$, $i = 1,\ldots n$;
2) Для кожного хешу $h_0, \ldots h_n$:

a) Створення схеми $CH_i(xH_i, wH_i)$ геш-алгоритму H, де відкритий вхід $xH_i = h_i$ є результатом хешування, свідок wH_i є прообразом хешу: $wH_0 = 0$, $wH_i = h_{i-1}||i$, $i = 1,...n$;

b) Генерація загальнодоступних параметрів $(SH_{pi}, SH_{vi}) = S(CH_i(xH_i, wH_i))$, де SH_{pi} — параметри загальнодоступного прувера, SH_{vi} — параметри публічного верифікатора;

c) Формування доказу ОЦ для хешування $\pi H_i = P(SH_{pi}, xH_i, wH_i)$;

d) Реалізація алгоритму перевірки реалізації $V(SH_{vi}, xH_i, \pi H_i)$, що приймає значення *{0, 1} (accept or reject)*;

e) Перевірка доказу, що $V(SH_{vi}, xH_i, \pi H_i) = accept$.

3) Для кожного підпису $EDS_i = E(h_i, sk)$, $i = 0,...n$:

a) Створення схеми $CD_i(xD_i, wD_i)$ перевірки підпису $h_i = D(EDS_i, pk)$, де відкритий вхід $xD_i = h_i$ є результатом хешування, свідок $wD_i = (EDS_i, pk)$ є підписом і відкритим ключем ;

b) Генерація загальнодоступних параметрів $(SD_{pi}, SD_{vi}) = S(CD_i(xD_i, wD_i))$, де SD_{pi} – параметри загальнодоступного верифікатора, SD_{vi} – параметри публічного верифікатора;

c) Формування доказу ОЦ для перевірки підпису $\pi D_i = P(SD_{pi}, xD_i, wD_i)$);

d) Реалізація алгоритму перевірки доказу $V(SD_{vi}, xD_i, \pi D_i)$, що приймає значення *{0, 1} (accept or reject)*;

e) Перевірка доказу, що $V(SD_{vi}, xD_i, \pi D_i) = accept$.

4) Для кожної трійки доказів $\prod_{i-1} = P(S_{Pi-1}, X_{i-1}, W_{i-1})$, $\pi H_i = P(SH_{pi}, xH_i, wH_i)$ and $\pi D_i = P(SD_{pi}, xD_i, wD_i)$, $i = 1,...n$:

a) Створення схеми $C_i(X_i, W_i)$ алгоритму перевірки V, де

i) $X_i = (V(S_{Vi-1}, X_{i-1}, \prod_{i-1})), V(SH_{vi}, xH_i, \pi H_i), V(SD_{vi}, xD_i, \pi D_i)$, для всіх $i = 1,...n$.

ii) $W_1 = (\pi_0, h_0, \pi_1, h_1, EDS_1, pk)$, $W_1 = (\prod_{i-1}, X_{i-1}, \pi_i, h_i, EDS_i, pk)$, для всіх $i = 2,...n$.

b) Генерація загальнодоступних параметрів $(S_{Pi}, S_{Vi}) = S(C(X_i, W_i))$, де S_{Pi} – параметри загальнодоступного прувера, S_{Vi} – параметри публічного верифікатора;

c) Формування доказу ОЦ $\prod_i = P(S_{Pi}, X_i, W_i)$;

d) Реалізація алгоритму перевірки доказів $V(S_{Vi}, X_i, \prod_i)$, що приймає значення *{0, 1} (accept or reject);*

e) Перевірка доказу, тобто переконатися, що $V(S_{Vi}, X_i, \prod_i) = accept$.

Таким чином, кожен доказ ∏i = P(SPi, Xi, Wi), i = 1,…n є сукупністю трьох інших доказів:

1) Доказ ОЦ попереднього ланцюжка пов'язаних хешів $\prod_{i-1} = P(S_{Pi-1}, X_{i-1}, W_{i-1})$;
2) Доказ ОЦ поточного хешу $\pi H_i = P(SH_{pi}, xH_i, wH_i)$;
3) Доказ ОЦ поточної перевірки підпису $\pi D_i = P(SD_{pi}, xD_i, wD_i)$.

Доведення $\prod_0 = P(S_{P0}, X_0, W_0)$ є сукупністю двох доказів:

1) Доказ ОЦ поточного хешу $\pi H_0 = P(SH_{p0}, xH_0, wH_0)$;
2) Доказ ОЦ поточної перевірки підпису $\pi D_0 = P(SD_{p0}, xD_0, wD_0)$.

Виконання умови $V(S_{Vi}, X_i, \prod_i) = accept$ для всіх $i = 1,…n$ означає, що перевірка доказу $\prod_{i-1} = P(S_{Pi-1}, X_{i-1}, W_{i-1})$, $\pi H_i = P(SH_{pi}, xH_i, wH_i)$ та $\pi D_i = P(SD_{pi}, xD_i, wD_i)$ обчислено правильно. Якщо $V(S_{Vi-1}, X_{i-1}, \prod_{i-1}) = accept$, $V(SH_{vi}, xH_i, \pi H_i) = accept$ та $V(SD_{vi}, xD_i, \pi D_i) = accept$, це означає, що:

- Існує доказ ОЦ попереднього ланцюга, тобто перевірка $V(S_{Vi-2}, X_{i-2}, \prod_{i-2}) = accept$ обчислена правильно;
- Є підтвердження ОЦ поточного хешу, тобто значення $h_i = H(h_{i-1}||i)$ обчислено правильно;
- Існує підтвердження ОЦ поточного підпису, тобто перевірка $h_i = D(EDS_i, pk)$ обчислена правильно.

Рекурсія. Для реалізації рекурсивного доказу ОЦ ланцюга необхідно послідовно виконувати наступні кроки:

1) Реалізація ланцюга блоків на основі пов'язаних хешів $h_0 = H(0)$, $h_i = H(h_{i-1}||i)$, $i = 1,…n$;

2) Для кожного блоку створення схемуи $C_i(X_i, W_i)$ геш-алгоритму H, перевірки підпису та алгоритму перевірки V для попереднього блоку $\pi_{i-1} = P(S_{pi-1}, x_{i-1}, w_{i-1})$. $X_i = (V(S_{vi-1}, x_{i-1}, \pi_{i-1}), x_i)$: $V(S_{vi-1}, x_{i-1}, \pi_{i-1})$ є результатом перевірки (accept or reject) попереднього доказу π_{i-1}, $x_i = h_i$ — результат хешування. $W_i = (\pi_{i-1}, h_{i-1}, wH_i, wD_i)$: π_{i-1}, h_{i-1} — вхідні дані для перевірки доказу, а w_i — вхідні дані (свідок, хеш-прообраз) для обчислення хешів: $w_0 = 0$, $w_i = (h_{i-1}||i)$, $i = 1,...n$, свідок $wD_i = (EDS_i, pk)$ – підпис і відкритий ключ;
3) Генерація загальнодоступних параметрів $(S_{pi}, S_{vi}) = S(CH_i(xH_i, wH_i), CD_i(xD_i, wD_i))$, де S_{pi} – публічні параметри перевірки, S_{vi} – загальнодоступні параметри перевірки;
4) Реалізація алгоритму перевірки доказів $V(S_{vi}, x_i, \pi_i)$ приймає значення *{0, 1} (accept or reject)*;
5) Перевірка доказу, тобто переконатися, що $V(S_{vi}, x_i, \pi_i) = accept$.

Виконання умови $V(S_{Vi}, X_i, \prod_i) = accept$ для всіх $i = 1,...n$ означає, що є доказ ОЦ попереднього ланцюжка, тобто перевірка $V(S_{vi-1}, x_{i-1}, \pi_{i-1}) = accept$ обчислена правильно, ОЦ поточного хешу, тобто значення $h_i = H(h_{i-1}||i)$ обчислено правильно, а ОЦ поточного підпису, тобто перевірка $h_i = D(EDS_i, pk)$ обчислена правильно. Схема наведена на рисунку додатку Т.

4.4 Експериментальне дослідження програмної реалізації системи перевірки валідності ланцюга блоків

У цьому розділі зробимо огляд реалізації програми, призначеної для підтвердження цілісності обчислень у ланцюжку взаємопов’язаних блоків.

В основі реалізації лежить фреймворк Plonky2 [81], широко визнаний інструмент у просторі блокчейн, розроблений компанією Polygon Zero [82]. Він відомий своєю універсальністю та ефективністю, що робить його ідеальним вибором для проектів, які потребують високого рівня безпеки та цілісності обчислень.

Plonky2 реалізує протокол PLONK, найсучаснішу систему доказів нульового знання в поєднанні з арифметизацією TurboPLONK – значне вдосконалення, яке оптимізує ефективність і продуктивність системи перевірки. В якості схеми зобов'язань Plonky2 використовує FRI, що підвищує безпеку та можливість перевірки системи доказів, надаючи надійний механізм перевірки цілісності обчислень у блокчейн.

Метою цього дослідження є внесення цінної інформації в постійну еволюцію безпечних і масштабованих блокчейн-рішень.

В якості прикладу для тестової мережі блокчейн було обрано хеш функцію SHA256 та протокол підпису ED25519 [83]. SHA-256 – криптографічна хеш-функція, що належить до сімейства SHA-2. Вона широко використовується в технології блокчейн та інших програмах безпеки. SHA-256 призначена для отримання вхідних даних (або повідомлень) будь-якої довжини та створення вихідних даних фіксованого розміру, довжиною 256 біт. Це одностороння функція, тобто обчислювально неможливо повернути процес назад і отримати вихідні дані з його хешу. ED25519 – алгоритм цифрового підпису еліптичної кривої на основі EdDSA (алгоритм цифрового підпису кривої Едвардса). ED25519 розроблений для високої безпеки та продуктивності, використовуючи криву Едвардса Curve25519. Відкриті ключі, згенеровані ED25519, мають довжину 256 біт, що забезпечує хороший баланс між безпекою та обчислювальною ефективністю. Підписи, створені ED25519, мають довжину 512 біт, що забезпечує безпечний засіб автентифікації.

ED25519 використовує криптографічну хеш-функцію SHA-512 для обробки повідомлень і створення цифрових підписів. Хеш-функція сприяє безпеці алгоритму, створюючи вихідні дані фіксованого розміру.

Отже, двома ключовими компонентами схеми доказу ланцюга пов’язанних блоків виступають схема перевірки валідності хешу на основі SHA-256 та підпису ED25519. Крім того, підпис використовує SHA-512, доказ якої також необхідно реалізувати.

У додатку У в таблицях У.1-У.3 наведено код, що генерує докази для кожного з алгоритмів та виконує їх перевірку.

Дану реалізацію було протестовано на ланцюгу із п’яти блоків для аналізу часових витрат та розміру фіаних доказів. Тестування було проведено на комп’ютері 1,900GHz AMD Ryzen 7 5800U (16).

У таблицях Ф.1-Ф.5 додатку Ф наведено результати для схеми агрегації та рекурсії.

На графіку нижче наведено результати обчислення часу для нативної перевірки (або перерахунку всіх хешів), перевірки доказів, згенерованих для кожного блоку, і рекурсивного доказу.

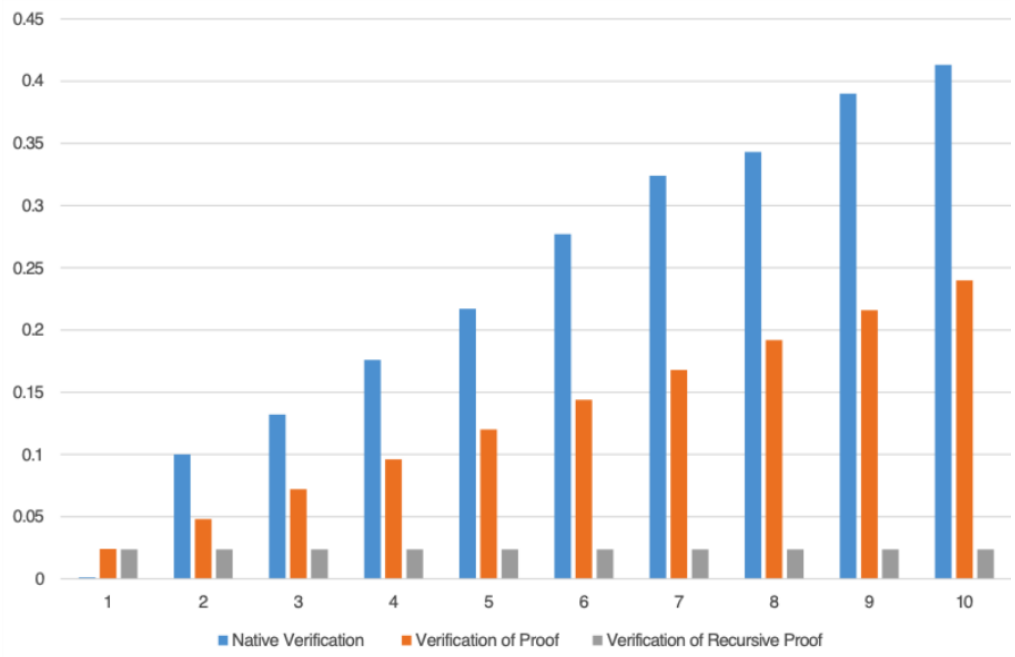

Рисунок 4.9 — Обчислювальна складність нативної перевірки, перевірки доказів, рекурсивної перевірки доказів

Бачимо, що навіть доказ для окремого блоку значно здешевлює перевірку. Рекурсія поєднує всі ці докази в одне. Верифікація дуже швидка, що, власне, і вирішує основну проблему масового впровадження розподілених систем – масштабованість.

Щоб бути більш точним, зазначимо, що в обчислювальних процесах могли статися системні помилки. Системні помилки під час обчислювальних процесів можуть виникати з різних джерел, що призводить до неточностей в

результатах. Крім того, виникають помилки округлення, коли числа з нескінченною кількістю десяткових розрядів наближаються до кінцевого представлення, вносячи розбіжності.

Незважаючи на це, експеримент демонструє, що докази з нульовим знанням є сильним рішенням для вирішення проблем масштабованості та конфіденційності. Це особливо важливо у великих розподілених обчислювальних проектах. Запроваджена система доказів може замінити нативну перевірку. Це значно прискорює верифікацію та полегшує роботу системи. Наведені часові та вимірювальні оцінки показують перспективність цього напрямку. На нашу думку, необхідно продовжувати дослідження цієї технології та впровадити системи перевірки ланцюга блоків на її основі.

ВИСНОВОК

Сучасні блокчейн-системи стикаються з проблемою масштабованості, що стосується підвищення спроможності мережі блокчейн обробляти зростаючу кількість транзакцій і користувачів без зниження продуктивності. Проблема масштабованості виникає через властиві компроміси між децентралізацією, безпекою та масштабованістю, відомі як «трилема блокчейн».

Було розглянуто різні напрямки, які намагаються знайти золоту середину між цими властивостями і зробити системи більш гнучкими, не втрачаючи при цьому конфіденційності та безпеки даних. Аналіз показав, що одним з найбільш перспективних підходів за своїми потужними характеристиками було обрано напрям доказу з нульовим знанням.

Технологія ZKP відіграє ключову роль у підвищенні конфіденційності та безпеки в децентралізованих мережах. Вона підвищує цілісність систем, одночасно захищаючи ідентифікаційні дані користувачів і деталі транзакцій. Користувачі мають змогу підтверджувати правдивість заяв, транзакцій та ідентифікаційних даних без необхідності розкривати додаткову інформацію. ZKP також дають змогу спільно аналізувати дані без шкоди для цілісності базових даних у сфері безпечних багатосторонніх обчислень. Це особливо важливо для сценаріїв, коли декільком сторонам потрібно спільно обробляти конфіденційні дані. Це створює потужний бар'єр проти невиправданого втручання в особисте життя та забезпечує надійний захист, необхідний для боротьби з кіберзлочинами, пов’язаними з особистими даними, забезпечуючи впевненість користувачів у безпеці їхніх ідентифікаційних даних в Інтернеті.

Крім того, ZKP спрощують процес адаптації на децентралізованих платформах. Користувачі можуть швидко й ефективно встановити свою цифрову ідентифікацію без громіздкого завдання надання великої кількості персональних даних. Тобто, технологія ZKP дотримується принципу мінімізації даних. Розкриваючи лише те, що є суттєвим для перевірки, вона значно

зменшує кількість даних, що перебувають у обігу. Це зменшення є ключовим фактором у зміцненні безпеки, оскільки менше інформації піддається потенційним атакам.

Таким чином, технологія Zero-Knowledge Proofs є передовою у трансформації методів перевірки ідентифікаційних даних у децентралізованих мережах. Вона надає користувачам безпечні механізми, спрямовані на збереження конфіденційності, та ефективні засоби адаптації до цифрових платформ, одночасно знижуючи ризики, пов'язані з крадіжкою особистих даних та шахрайством. З впровадженням децентралізованих мереж, роль технології ZKP в збереженні цілісності та конфіденційності цифрових ідентифікаційних даних стає ще більш суттєвою, забезпечуючи безперервний розвиток безпечної цифрової екосистеми.

У ході дослідження було проаналізовано всі існуючі та зарекомендовані у блокчейн ком’юніті типи доказів з нульовим знанням, На нашу думку, необхідно привернути увагу протоколу PLONK та схемі зобов’язань FRI.

PLONK – це протокол доказу нульового знання, що реалізує модель ZK-SNARK та FRI, що є квантово стійким і не потребує надійних налаштувань, відіграють вирішальну роль у вирішенні проблем масштабованості блокчейн. PLONK і FRI оптимізують процес перевірки, дозволяючи мережам ефективно обробляти великі обсяги транзакцій, що має вирішальне значення для масштабованості мереж блокчейн. Крім того, вони покращують конфіденційність, дозволяючи перевіряти транзакції без розголошення таємної інформації. Властивість нульового знання гарантує, що для перевірки будуть розкриті лише необхідні деталі, зберігаючи конфіденційність користувача.

PLONK і FRI використовуються у рішеннях другого рівня для покращення масштабованості. Переносячи обробку транзакцій на другий рівень, навантаження на основний блокчейн зменшується, вирішуючи проблеми масштабованості. PLONK і FRI отримали впровадження в різних блокчейн-проектах, демонструючи свою універсальність і ефективність у підвищенні масштабованості.

Аналіз дослідження деталей наведених протоколів показав, що вони є перспективним напрямком у вирішенні проблеми масштабованості блокчейн мереж в різних застосунках.

Крім того, в ході роботи було розроблено схему доказу валідності ланцюга блоків із застосуванням обраних протоколів. Впродовж експерименту було відмічено суттєве зниження витрат на перевірку блоків із запровадженням доказів для кожного блоку. Рекурсія консолідує ці докази, що забезпечує швидку перевірку всього ланцюга. Це ефективно вирішує основну проблему масштабованості в умовах широкого впровадження розподілених систем. Більш того, проведений експеримент підкреслює, що докази з нульовим знанням пропонують відмінне вирішення проблем конфіденційності, особливо в масштабних розподілених обчислювальних проектах.

Отже, запроваджена система перевірки здатна замінити звичайні методи перевірки, значно прискорюючи процес підтвердження валідності ланцюга блоків та підвищуючи простоту експлуатації технології.

Надані оцінки часу та вимірювань підкреслюють багатообіцяючі перспективи в напрямку доказів із нульовим знанням. З нашої точки зору, продовження досліджень цієї технології є перспективним науковим напрямком. Це, у свою чергу, вимагає розробки та впровадження систем перевірки блокчейн на основі доказів з нульовим знанням.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. NIST. Blockchain Technology Overview. URL: https://nvlpubs.nist.gov/nistpubs/ir/2018/nist.ir.8202.pdf (дата звернення: 8.05.2023)
2. Структура Меркла-Дамгора. URL: https://ru.wikipedia.org/wiki/Структура_Меркла_—_Дамгора (дата звернення: 12.05.2023)
3. The History & Future of Blockchain Technology. URL: https://www.linkedin.com/pulse/history-future-blockchain-technology-the-coin-times (дата звернення: 31.05.2023)
4. Dawit Andargachew Asmare. *Blockchain Technology: Understanding its Meaning, Architecture, and Diverse Applications.* URL: https://www.researchgate.net/figure/Stuart-Haber-and-W-Scott-Stornetta_fig3_373236964 (дата звернення: 31.05.2023)
5. W. Diffie, M. E. Hellman. *New Directions in Cryptography.* URL: https://www-ee.stanford.edu/~hellman/publications/24.pdf (дата звернення: 2.06.2023)
6. J. Pan al. *Signed (Group) Diffie-Hellman Key Exchange with Tight Security.* URL: https://eprint.iacr.org/2021/607.pdf (дата звернення: 10.06.2023)
7. S. Haber, W. Stornetta. *How To Time-Stamp a Digital Document.* URL: https://link.springer.com/content/pdf/10.1007/BF00196791.pdf (дата звернення: 15.07.2023)
8. L. Lamport. *Paxos Made Simple.* URL: https://lamport.azurewebsites.net/pubs/paxos-simple.pdf (дата звернення: 15.07.2023)
9. D. Ongaro, J. Ousterhout. *In Search of an Understandable Consensus Algorithm (Extended Version).* URL: https://raft.github.io/raft.pdf (дата звернення: 15.07.2023)

10. M. Castro, B. Liskov. *Practical Byzantine Fault Tolerance.* URL: http://pmg.csail.mit.edu/papers/osdi99.pdf (дата звернення: 8.08.2023)
11. BitTorrent. URL: https://www.bittorrent.com (дата звернення: 8.08.2023)
12. Consensus Algorithms. URL: https://aeroncookbook.com/distributed-systems-basics/consensus-algorithms/ (дата звернення: 8.08.2023)
13. Gnutella. URL: https://en.wikipedia.org/wiki/Gnutella (дата звернення: 8.08.2023)
14. Bircoin. URL: https://bitcoin.org/en/ (дата звернення: 8.08.2023)
15. S. Nakamoto. *Bitcoin: A peer-to-peer electronic cash system.* https://bitcoin.org/bitcoin.pdf (дата звернення: 20.08.2023)
16. Ethereum. URL: https://ethereum.org/en/ (дата звернення: 20.08.2023)
17. Solana. URL: https://solana.com (дата звернення: 20.08.2023)
18. BNB. URL: https://www.binance.com/en/bnb (дата звернення: 20.08.2023)
19. J. Frankenfield. *Cryptocurrency Explained With Pros and Cons for Investment.* URL: https://www.investopedia.com/terms/c/cryptocurrency.asp (дата звернення: 22.08.2023)
20. Blockchain Scalability: Exploring Solutions in Blockchain Space. URL: https://www.linkedin.com/pulse/blockchain-scalability-exploring-solutions (дата звернення: 22.08.2023)
21. D. Kraus, T. Obrist, O. Hari. *Blockchains, Smart Contracts, Decentralised Autonomous Organisations and the Law.* URL: https://www.google.it/books/edition/Blockchains_Smart_Contracts_Decentralise/ui6UDwAAQBAJ?hl=ru&gbpv=1&dq=blockchain+in+smart+contracts&pg=PA114&printsec=frontcover (дата звернення: 13.08.2023)
22. Smart contract. URL: https://en.wikipedia.org/wiki/Smart_contract (дата звернення: 24.08.2023)
23. U. W. Chohan. *Non-Fungible Tokens: Blockchains, Scarcity, and Value.* URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3822743 (дата звернення: 24.08.2023)

24. NFT. URL: https://www.theverge.com/22310188/nft-explainer-what-is-blockchain-crypto-art-faq (дата звернення: 25.08.2023)
25. OpenSea. URL: https://opensea.io/ (дата звернення: 25.08.2023)
26. Rarible. URL: https://rarible.com/ (дата звернення: 25.08.2023)
27. I. Ershov. *Blockchain gaming: An analysis of the use of blockchain technology in the video gaming industry.* URL: https://amslaurea.unibo.it/20533/1/ershov_igor_tesi.pdf (дата звернення: 18.08.2023)
28. What are NFT games? URL: https://blog.chain.link/nft-games/ (дата звернення: 28.08.2023)
29. CryptoKitties. URL: https://ru.wikipedia.org/wiki/CryptoKitties (дата звернення: 28.08.2023)
30. DLT. URL: https://www.ibm.com/cloud/learn/blockchain-distributed-ledgers (дата звернення: 28.08.2023)
31. What Is Decentralization In Blockchain? URL: https://www.blockchain-council.org/blockchain/what-is-decentralization-in-blockchain/ (дата звернення: 28.08.2023)
32. K.Wust, A. Gervais. *Do you need a Blockchain?* URL: https://eprint.iacr.org/2017/375.pdf (дата звернення: 15.09.2023)
33. Blockchain and Data Integrity in Computer System Validation. URL: https://fivevalidation.com/blockchain-and-data-integrity-in-computer-system-validation/ (дата звернення: 15.09.2023)
34. Cryptographic hash functions. URL: https://en.wikipedia.org/wiki/Cryptographic_hash_function (дата звернення: 4.09.2023)
35. SHA-2. URL: https://ru.wikipedia.org/wiki/SHA-2 (дата звернення: 15.09.2023)
36. SHA-3. URL: https://en.wikipedia.org/wiki/SHA-3 (дата звернення: 15.09.2023)

37. RIPEMD. URL: https://en.wikipedia.org/wiki/RIPEMD (дата звернення: 15.09.2023)
38. V. Buterin. *Why sharding is great: demystifying the technical properties*. URL: https://vitalik.ca/general/2021/04/07/sharding.html (дата звернення: 1.09.2023)
39. BTC & ETH Transactions: веб-сайт. URL: https://bitinfocharts.com/comparison/transactions-btc-eth.html#alltime (дата звернення: 16.09.2023)
40. K. R. W. Meijer. *Blockchain and the scalability challenge: solving the blockchain trilemma.* URL: https://www.finextra.com/blogposting/24941/blockchain-and-the-scalability-challenge-solving-the-blockchain-trilemma (дата звернення: 25.08.2023)
41. Blockchain Scalability: Challenges and Recent Developments. URL: https://www.codementor.io/blog/blockchain-scalability-5rs5ra8eej (дата звернення: 16.08.2023)
42. What Are The Blockchain Scalability Issues And How Can They Be Solved? URL: https://www.doubloin.com/learn/blockchain-scalability-issues (дата звернення: 16.09.2023)
43. Blockchain Verification Process: Explained. URL: https://www.solulab.com/how-blockchain-verification-work/ (дата звернення: 1.10.2023)
44. C. Barrera. *Hidden Costs of Verification.* URL: https://medium.com/mit-cryptoeconomics-lab/hidden-costs-of-verification-4925cac38a47 (дата звернення: 2.09.2023)
45. Ethereum. Plasma. URL: https://ethereum.org/en/developers/docs/scaling/plasma/ (дата звернення: 5.09.2023)
46. What are blockchain rollups? URL: https://www.ledger.com/academy/what-are-blockchain-rollups (дата звернення: 5.09.2023)

47. Ethereum. State channels. URL: https://ethereum.org/en/developers/docs/scaling/state-channels/ (дата звернення: 5.09.2023)
48. What is a side channel. URL: https://www.ledger.com/academy/what-is-a-sidechain (дата звернення: 7.09.2023)
49. Lightning network. URL: https://lightning.network (дата звернення: 12.10.2023)
50. Raiden. URL: https://raiden.network (дата звернення: 12.10.2023)
51. Polkadot. URL: https://polkadot.network (дата звернення: 12.10.2023)
52. Cosmos. URL: https://cosmos.network (дата звернення: 13.10.2023)
53. Ethereum roadmap. Ethereum 2.0. URL: https://ethereum.org/en/roadmap/ (дата звернення: 15.10.2023)
54. Tezos. URL: https://tezos.com/proof-of-stake/ (дата звернення: 26.10.2023)
55. SegWit. URL: https://en.wikipedia.org/wiki/SegWit (дата звернення: 26.10.2023)
56. Zero-Knowledge proofs. URL: https://en.wikipedia.org/wiki/Zero-knowledge_proof (дата звернення: 6.10.2023)
57. Zero-knowledge proofs – a powerful addition to blockchain. URL: https://blockheadtechnologies.com/zero-knowledge-proofs-a-powerful-addition-to-blockchain/ (дата звернення: 6.10.2023)
58. Cryptographic Proof Systems. URL: https://www.cs.utexas.edu/~dwu4/proofs-project.html (дата звернення: 7.10.2023)
59. Cambrian Explosion of Cryptographic Proofs. URL: https://medium.com/starkware/cambrian-explosion-of-cryptographic-proofs-5740a41cdbd2 (дата звернення: 7.10.2023)
60. Aztec. URL: https://aztec.network (дата звернення: 8.10.2023)

61. Starknet. URL: https://www.starknet.io/en (дата звернення: 8.10.2023)
62. Understanding Zero-Knowledge Proofs in 15 Mins through SNARK and STARK. URL: https://intelchen.medium.com/understanding-zero-knowledge-proofs-in-15-mins-through-snark-and-stark-7638311f0cc9 (дата звернення: 4.10.2023)
63. Ethereum. Zero-Knowledge proofs. URL: https://ethereum.org/en/zero-knowledge-proofs/#what-are-zk-proofs (дата звернення: 2.10.2023)
64. Schnorr & Sigma protocols. URL: https://crypto.sjtu.edu.cn/~yandi/2018%20BIU%20winter%20school/Part%203-Techniques%20for%20Efficient%20ZK%20(cont.)/WS-19-11-sigma-protocols-winter-school-2019-1.pdf (дата звернення: 12.09.2023)
65. A. Gonz'alez, C. R'afols. *New Techniques for Non-Interactive Shuffle and Range Arguments.* URL: https://repositori.upf.edu/bitstream/handle/10230/46991/rafols_ACNS2016_new.pdf?sequence=1 (дата звернення: 6.10.2023)
66. B. Bunz, J. Bootle, Dan Boneh. *Bulletproofs: Short Proofs for Confidential Transactions and More.* URL: https://eprint.iacr.org/2017/1066.pdf (дата звернення: 16.10.2023)
67. Non-interactive zero-knowledge proof. URL: https://en.wikipedia.org/wiki/Non-interactive_zero-knowledge_proof (дата звернення: 18.10.2023)
68. Eli Ben-Sasson, Alessandro Chiesa. *Succinct Non-Interactive Zero Knowledge for a von Neumann Architecture.* URL: https://eprint.iacr.org/2013/879.pdf (дата звернення: 20.10.2023)
69. What Is ZK-STARK? URL: https://starkware.co/stark/ (дата звернення: 30.09.2023)
70. Arithmetization. URL: https://medium.com/starkware/arithmetization-i-15c046390862 (дата звернення: 15.10.2023)

71. U. Habock. *A summary on the FRI low degree test.* URL: https://eprint.iacr.org/2022/1216.pdf (дата звернення: 28.10.2023)
72. Aniket Kate, Gregory M. Zaverucha, Ian Goldberg. *Constant-Size Commitments to Polynomials and Their Applications?* URL: https://www.iacr.org/archive/asiacrypt2010/6477178/6477178.pdf (дата звернення: 16.10.2023)
73. Comparison of Different ZK-SNARKs. URL: https://medium.com/@daniel.linfeng.zhou/comparison-of-different-zk-snarks-3f3ac7dd8a4a (дата звернення: 2.11.2023)
74. Comparing General Purpose ZK-SNARKs. URL: https://medium.com/coinmonks/comparing-general-purpose-zk-snarks-51ce124c60bd (дата звернення: 2.11.2023)
75. A. Gabizon, Z. J. Williamson, O. Ciobotaru. *PlonK: Permutations over Lagrange-bases for Oecumenical Non-interactive arguments of Knowledge.* URL: https://eprint.iacr.org/2019/953.pdf (дата звернення: 28.10.2023)
76. PLONK Benchmarks I — 2.5x faster than Groth16 on MiMC. URL: https://medium.com/aztec-protocol/plonk-benchmarks-2-5x-faster-than-groth16-on-mimc-9e1009f96dfe (дата звернення: 3.11.2023)
77. On the Optimization of PLONK. URL: https://zeroknowledge.fm/news-2-on-optimization-plonk/ (дата звернення: 16.10.2023)
78. V. Buterin. *QAP.* URL: https://medium.com/@VitalikButerin/quadratic-arithmetic-programs-from-zero-to-hero-f6d558cea649 (дата звернення: 6.10.2023)
79. V. Buterin. *Understanding PLONK.* URL: https://vitalik.ca/general/2019/09/22/plonk.html (дата звернення: 5.10.2023)
80. V. Buterin. *STARKs, Part II: Thank Goodness It's FRI-day.* URL: https://vitalik.ca/general/2017/11/22/starks_part_2.html (дата звернення: 3.10.2023)

81. Plonky2. URL: https://github.com/0xPolygonZero/plonky2/tree/main (дата звернення: 9.10.2023)
82. Polygon Zero. URL: https://polygon.technology/blog/polygon-announces-the-worlds-first-zero-knowledge-zk-scaling-solution-fully-compatible-with-ethereum (дата звернення: 10.10.2023)
83. EdDSA. URL: https://ru.wikipedia.org/wiki/EdDSA (дата звернення: 11.09.2023)

ДОДАТОК А

Таблиця А.1 – Виклики безпеки, децентралізації та масштабованості

Проблеми безпеки	Атаки 51%. блокчейн на основі PoW чутливі до атак 51%, коли один об’єкт або група майнерів, що володіють понад 50% обчислювальної потужності мережі, можуть маніпулювати історією транзакцій блокчейна.
	Вразливості смарт-контрактів. Розумні контракти, ключова функція багатьох блокчейн, схильні до помилок кодування та вразливостей, якими можуть скористатися зловмисники, що призведе до значних фінансових втрат.
	Керування особистими ключами. Приватні ключі користувачів мають вирішальне значення для захисту їхніх активів у блокчейн. Якщо приватні ключі втрачено або зламано, користувачі можуть втратити доступ до своїх коштів.

Продовження таблиці А.1 – Виклики безпеки, децентралізації та масштабованості

Проблеми децентралізації	Централізація майнінгу. У блокчейнах PoW потужність майнінгу часто зосереджена в руках кількох великих майнінг-пулів або організацій, що викликає занепокоєння щодо централізації та контролю.
	Обмежена різноманітність вузлів. Кількість повних вузлів у деяких блокчейнах відносно невелика, що може призвести до централізації та зниження стійкості мережі.
	Регуляторний тиск. Регуляторні органи в різних країнах можуть накладати обмеження на блокчейн і криптовалютну діяльність, що потенційно може призвести до централізації, оскільки учасники прагнуть дотримуватися правил.

Продовження таблиці А.1 – Виклики безпеки, децентралізації та масштабованості

Проблеми масштабованості	Обмежена пропускна здатність транзакцій. Багато популярних блокчейн: Bitcoin, Ethereum тощо – мають проблеми з обмеженими можливостями обробки транзакцій. Це призводить до довшого часу підтвердження та вищих комісій у періоди високого попиту.
	Ресурсомісткість. Блокчейн-мережі, такі як Bitcoin, які використовують консенсус Proof of Work, вимагають значної обчислювальної потужності та споживання енергії. Це робить їх менш масштабованими та менш екологічними.
	Розмір блокчейн. У міру того як реєстр блокчейн зростає, новим вузлам стає дедалі складніше приєднатися до мережі та підтримувати копію всієї книги, потенційно централізуючи контроль.

ДОДАТОК Б

Таблиця Б.1 – Переваги та недоліки рішень другого рівня

№	Переваги	Недоліки
1.	Масштабованість. Основною перевагою рішень другого рівня є їх здатність значно підвищувати пропускну здатність транзакцій, дозволяючи здійснювати транзакції швидше та ефективніше. Це особливо важливо для блокчейн-мереж, які стикаються з проблемами масштабованості.	Ризики безпеки. Користувачі несуть відповідальність за безпеку своїх рішень. Неправильне керування може призвести до втрати коштів. Ця додаткова відповідальність збільшує складність впровадження.
2.	Конфіденційність. Взаємодії поза мережею в рамках рішень другого рівня забезпечують підвищену конфіденційність порівняно з транзакціями в мережі, оскільки більшість деталей залишаються поза межами загальнодоступного блокчейн.	Проблеми централізації. Централізація може викликати занепокоєння на ранніх етапах впровадження другого рівня, оскільки деякі вузли або канали можуть мати більший потенціал або вплив, ніж інші.

Продовження таблиці Б.1 – Переваги та недоліки рішень другого рівня

№	Переваги	Недоліки
3.	Зменшене навантаження на блокчейн. Здійснюючи численні транзакції поза мережею, рішення другого рівня зменшують перевантаження основного блокчейн, зберігаючи його безпеку та децентралізацію.	Складність маршрутизації. Маршрутизація платежів між учасниками мережі каналів другого рівня може бути складною, особливо для великих транзакцій або коли канали не мають достатньої пропускної здатності.
4.	Низькі витрати на транзакцію. Транзакції, які проводяться на другому рівні, часто пов’язані з нижчими комісіями порівняно з транзакціями в основній мережі, що робить мікротранзакції та платежі на невеликі суми економічно доцільними.	Складність. Налаштування та керування рішеннями другого рівня, включаючи платіжні канали та канали стану, може бути складним і незручним для звичайного користувача блокчейн. Ця складність може перешкодити впровадженню.
5.	Низька затримка. Рішення другого рівня забезпечують майже миттєві транзакції, що робить їх придатними для додатків у реальному часі, мікротранзакцій і послуг, які вимагають швидкого часу відповіді.	Обмежене застосування. Рішення другого рівня, хоча й багатообіцяючі, все ще перебувають на ранніх стадіях впровадження та можуть мати обмежену доступність для звичайних користувачів і програм.

ДОДАТОК В

Таблиця В.1 – Переваги та недоліки рішень поза мережею

№	Переваги	Недоліки
1.	Масштабованість. Зменшуючи кількість транзакцій, які необхідно обробляти в основному блокчейн, позаланцюгові механізми можуть значно збільшити пропускну здатність мережі.	Проблеми з безпекою. Безпека позамережевих транзакцій зазвичай залежить від криптографічних методів і дизайну смарт-контрактів. Користувачам може знадобитися заблокувати активи як заставу, вводячи нові міркування безпеки.
2.	Конфіденційність. Транзакції поза мережею можуть запропонувати підвищену конфіденційність, оскільки вони проводяться приватно між сторонами. Це може бути важливо для певних випадків використання, коли деталі транзакції повинні бути конфіденційними.	Доступність даних. Забезпечення доступності та цілісності даних для транзакцій поза мережею може бути складним завданням, адже Користувачі та валідатори повинні брати участь у механізмах підтримки безпеки даних.

Продовження таблиці В.1 – Переваги та недоліки рішень поза мережею

№	Переваги	Недоліки
3.	Швидкість. Транзакції поза ланцюгом можуть оброблятися набагато швидше, ніж транзакції в ланцюзі, оскільки їм не потрібно проходити через механізм консенсусу блокчейн. Це призводить до швидшого часу підтвердження та зменшення затримки.	Сумісність. Забезпечення безпроблемної передачі активів і даних між різними рішеннями поза мережею та основним блокчейном може бути технічно складним.
4.	Економічна ефективність. Рішення поза ланцюгом часто призводять до нижчих операційних витрат. Зменшення активності в ланцюжку означає менші комісії, що робить його можливим для мікротранзакцій та інших транзакцій з невеликою вартістю.	Прийняття. Широке впровадження позамережевих рішень і стандартів сумісності є важливим для їх успіху.

ДОДАТОК Г

Таблиця Г.1 – Переваги та недоліки рішень паралельного блокчейн

№	Переваги	Недоліки
1.	Масштабованість. Розподіляючи навантаження транзакцій між кількома ланцюжками, мережа може обробляти більший обсяг транзакцій одночасно.	Ризик безпеки. Безпека всієї екосистеми може залежати від безпеки основного блокчейн та з'єднань між паралельними блокчейн. Уразливості або атаки на один ланцюжок потенційно можуть вплинути на всю мережу.
2.	Швидкість. Транзакції, що проводяться на паралельних блокчейнах, можуть бути швидшими та ефективнішими, оскільки їм не потрібно проходити той самий процес консенсусу, що й основний блокчейн. Це робить паралельні блокчейн добре придатними для додатків, які потребують швидких транзакцій із малою затримкою.	Проблеми з ліквідністю. Користувачам може знадобитися заблокувати активи або токени в одному блокчейн, щоб отримати доступ до послуг або активів в іншому. Це може призвести до проблем з ліквідністю, особливо для користувачів з обмеженими коштами.

Продовження таблиці Г.1 – Переваги та недоліки рішень паралельного блокчейн

№	Переваги	Недоліки
3.	Налаштування. Кожен паралельний блокчейн можна налаштувати відповідно до конкретних випадків використання, дозволяючи розробникам вибирати консенсусні механізми та правила, які найкраще відповідають їхнім програмам.	Управління та координація. Управління управлінням і координацією кількох паралельних блокчейн може бути складним. Розбіжності чи суперечки між паралельними ланцюгами можуть вимагати ефективних механізмів вирішення.
4.	Нижчі витрати. Паралельні блокчейн можуть зменшити витрати, пов’язані з виконанням транзакцій і запуском смарт-контрактів, оскільки вони часто мають різні механізми консенсусу або структури управління, які можуть бути економічно ефективнішими.	Складність взаємодії. Забезпечення того, що дані й активи можна легко й безпечно передавати між різними ланцюгами, є проблемою, яка потребує ретельного проектування та розробки.

ДОДАТОК Д

Таблиця Д.1 – Переваги та недоліки шардингу

№	Переваги	Недоліки
1.	Масштабованість. Розділивши блокчейн на менші сегменти, кожен шард може обробляти свої транзакції та смарт-контракти незалежно. Паралелізація обробки значно збільшує загальну пропускну здатність мережі та зменшує перевантаження основного ланцюга.	Ризик безпеці. Якщо зламано один шард, це потенційно може вплинути на безпеку всієї мережі. Необхідні належні заходи безпеки та тестування.
2.	Конфіденційність. Розділення транзакції на різні шарди обмежує видимість транзакцій лише тими, хто бере участь в одному сегменті.	Механізми консенсусу. Для шардингу можуть знадобитися зміни в механізмах консенсусу, що може внести складність і потенційно вплинути на безпеку мережі.

Продовження таблиці Д.1 – Переваги та недоліки шардингу

№	Переваги	Недоліки
3.	Децентралізація. Шардинг може покращувати децентралізацію мережі, оскільки більше вузлів можуть брати участь у консенсусних процесах, а не покладатися на кілька централізованих вузлів для перевірки транзакцій.	Доступність даних. У деяких моделях сегментування не всі вузли зберігають усі дані сегментів, що може спричинити проблеми з доступністю даних.
4.	Ефективність. Шардинг значно підвищує енергоефективність, оскільки обчислювальні ресурси, необхідні для консенсусу та обробки транзакцій, розподіляються між кількома шардами.	Оновлення мережі. Реалізація сегментування часто потребує оновлень мережі, що може бути спірним та складним для досягнення, особливо якщо учасники мережі мають різні погляди на підхід.
5.	Швидші транзакції. Завдяки шардингу транзакції в шарді можна обробляти швидше, оскільки їм не потрібно чекати підтвердження в основному ланцюжку.	Проблеми сумісності. Хоча шардинг може покращити взаємодію в мережі, він може не вирішувати проблеми сумісності з іншими блокчейн-мережами. Міжланцюгова комунікація залишається складною проблемою.

ДОДАТОК Ж

Таблиця Ж.1 – Переваги та недоліки змін консенсусного алгоритму

№	Переваги	Недоліки
1.	Масштабованість. Нові механізми консенсусу можуть ефективніше обробляти транзакції, збільшуючи пропускну здатність мережі.	Мережеві форки. Зміна алгоритму консенсусу може призвести до мережевих форків, якщо між учасниками немає консенсусу, що потенційно може спричинити плутанину та фрагментацію блокчейн.
2.	Зменшена централізація. Деякі нові алгоритми консенсусу розроблені так, щоб бути більш децентралізованими та стійкими до концентрації влади, вирішуючи проблеми щодо централізації мережі.	Технічні проблеми. Впровадження нового механізму консенсусу може бути технічно складним і вимагати значної розробки та тестування.
3.	Покращення безпеки. Зміни консенсусного алгоритму можуть покращити безпеку, зробивши мережу більш стійкою проти атак.	Опір спільноти. Учасники та зацікавлені сторони, які інвестували в існуючий механізм консенсусу, можуть чинити опір змінам, що призведе до дебатів і суперечок у спільноті блокчейн.

Продовження таблиці Ж.1 – Переваги та недоліки змін консенсусного алгоритму

№	Переваги	Недоліки
4.	Швидша обробка транзакцій. Оновлені механізми консенсусу часто призводять до швидшого часу підтвердження транзакцій, покращуючи взаємодію з користувачем.	Проблеми взаємодії. Перехід до нового консенсусного алгоритму може спричинити проблеми сумісності з існуючими системами та програмами.

ДОДАТОК К

Таблиця К.1 – Переваги та недоліки SegWit

№	Переваги	Недоліки
1.	Покращена безпека. Відокремлення даних свідків покращує безпеку. Це пом’якшує певні типи податливості транзакцій, вразливість, яка потенційно може вплинути на стабільність і надійність мережі.	Проблеми впровадження. Незважаючи на те, що SegWit був широко прийнятий біткойн, його впровадження на інших блокчейн-платформах може бути більш складним і може вимагати консенсусу в усій мережі.
2.	Масштабованість. SegWit значно покращує масштабованість мережі блокчейн, збільшуючи кількість транзакцій, які можна обробити в межах блоку.	Довіра до впровадження. Якщо впровадження SegWit обмежене, покращення масштабованості можуть бути менш вираженими.
3.	Збільшена пропускна здатність транзакцій. Розділяючи дані, SegWit зменшує розмір окремих транзакцій. Це, збільшує кількість транзакцій, які можуть поміститися в один блок, підвищуючи загальну пропускну здатність блокчейн.	Складність. Впровадження SegWit може бути технічно складним і може вимагати оновлення гаманців і програмного забезпечення.

ДОДАТОК Л

Таблиця Л.1 – Переваги та недоліки доказів з нульовим знанням

№	Переваги	Недоліки
1.	Покращена конфіденційність. ZKP забезпечують надійний рівень конфіденційності, забезпечуючи конфіденційність транзакцій без шкоди для безпеки.	Проблеми з розміром перевірки. деякі впровадження ZKP можуть призвести до більших розмірів перевірки, впливаючи на ефективність і потенційно сповільнюючи процеси перевірки.
2.	Покращення масштабованості. увімкнувши перевірку поза мережею, ZKP усувають проблеми масштабованості, з якими стикаються традиційні системи блокчейн, підвищуючи пропускну здатність мережі.	Довіра до налаштування. Деякі протоколи потребують етап налаштування, який має проводитися чесно та безпечно, оскільки скомпрометований етап налаштування може створити вразливі місця.
3.	Інтероперабельність. ZKP сприяють взаємодії, безпечно сприяючи обміну даними між різними блокчейн-мережами.	Складність. Впровадження zk-SNARK у мережу блокчейн може потребує досвіду передової криптографії.

ДОДАТОК М

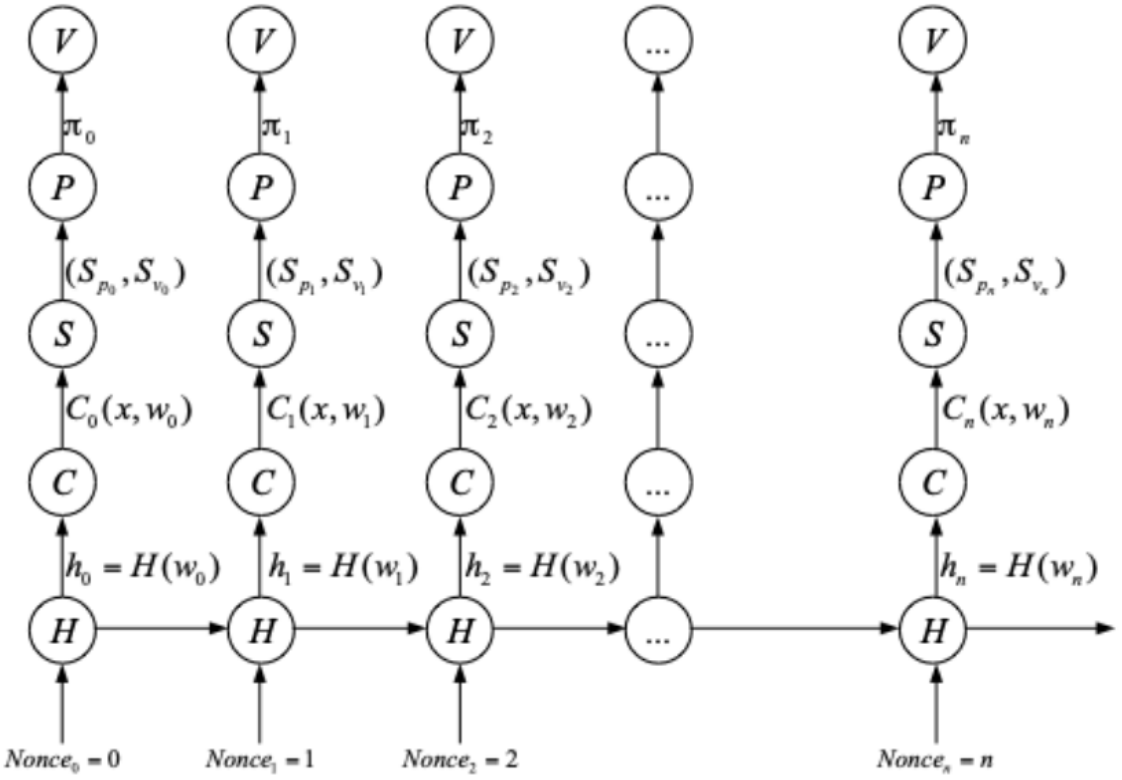

Рисунок М.1 — Схема генерації набору доказів обчислювальної цілісності

ДОДАТОК Н

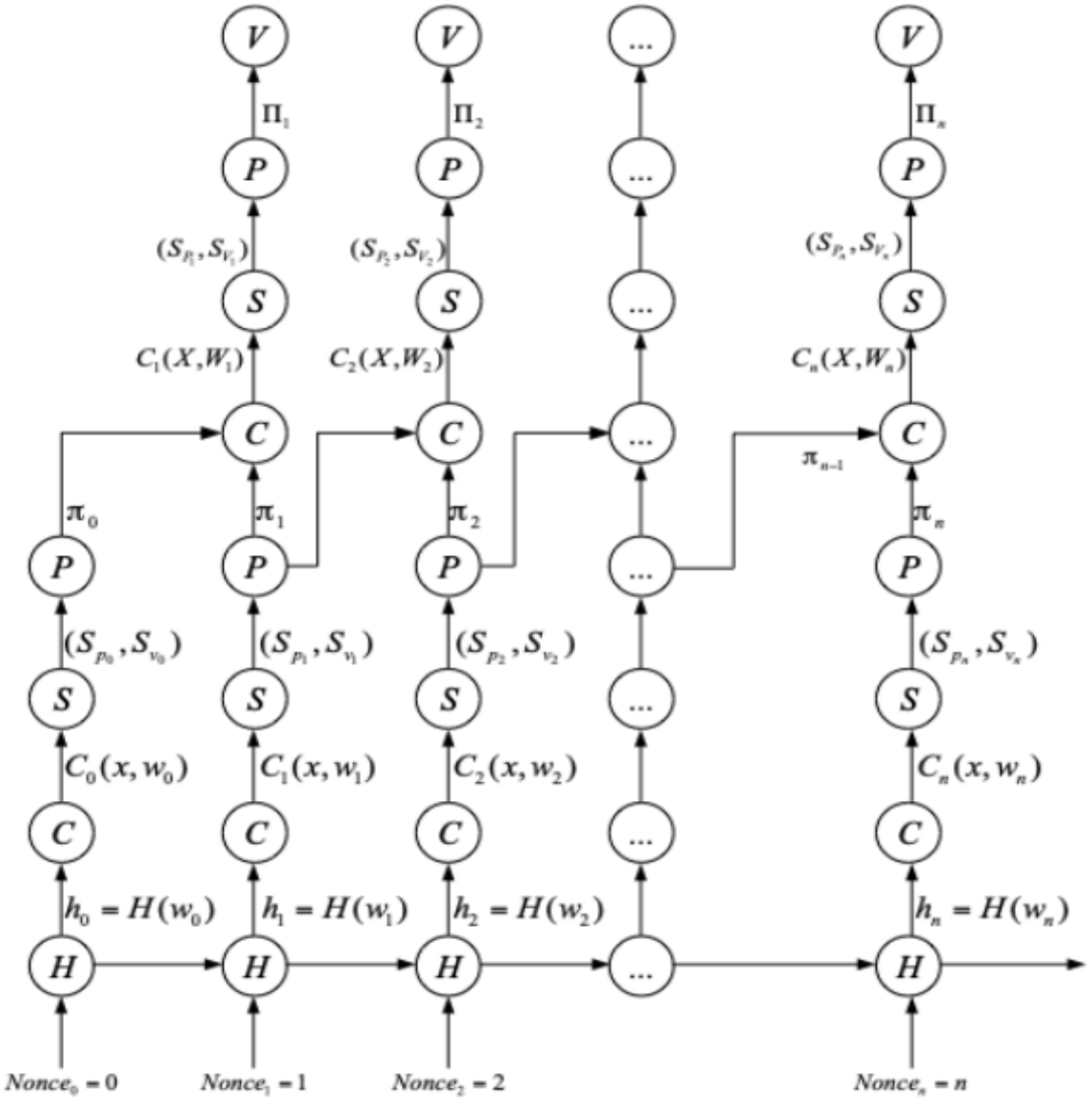

Рисунок Н.1 — Схема формування ланцюга рекурсивних доказів обчислювальної цілісності

ДОДАТОК П

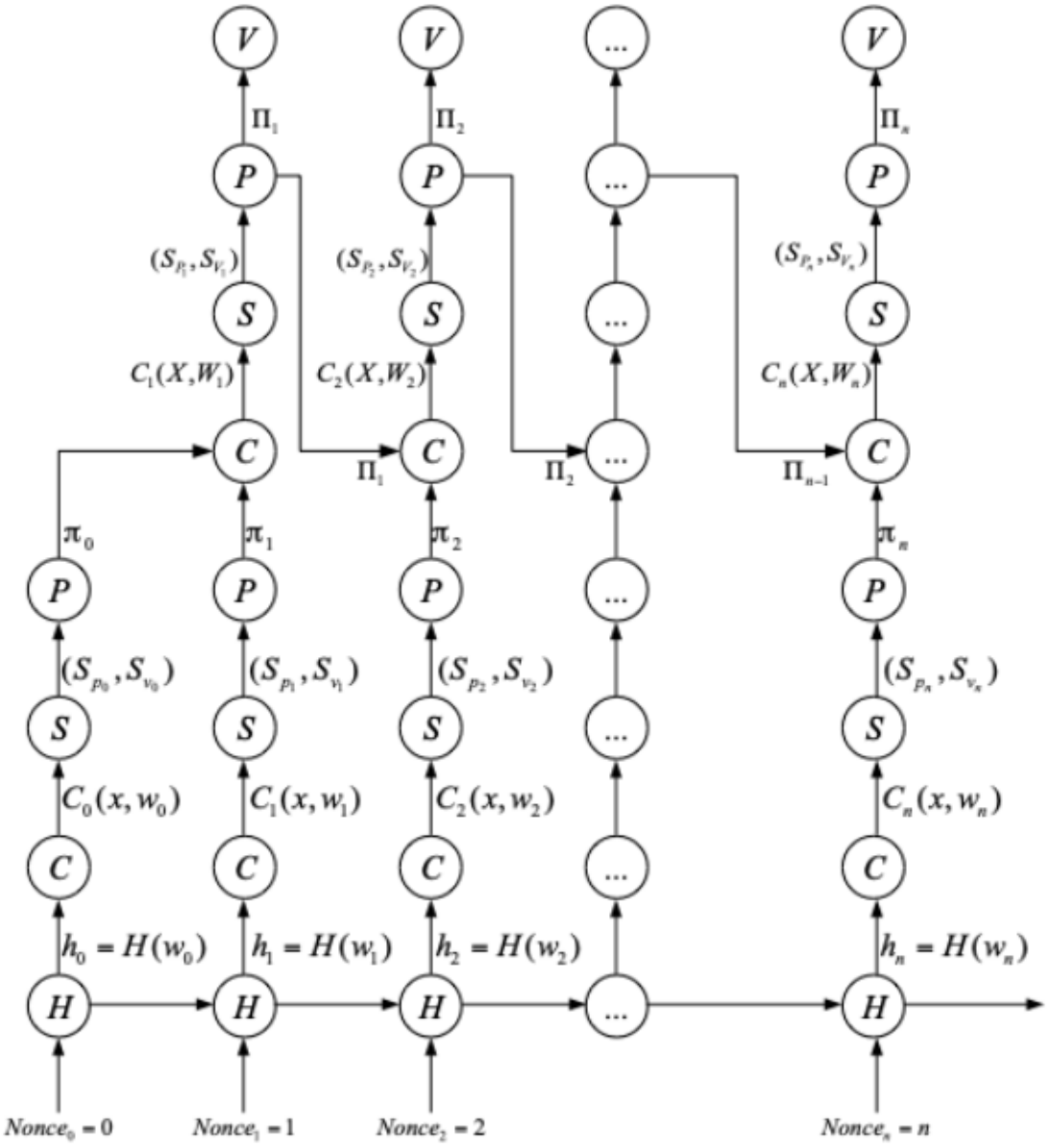

Рисунок П.1 — Схема формування ланцюжка рекурсивних доказів обчислювальної цілісності

ДОДАТОК Р

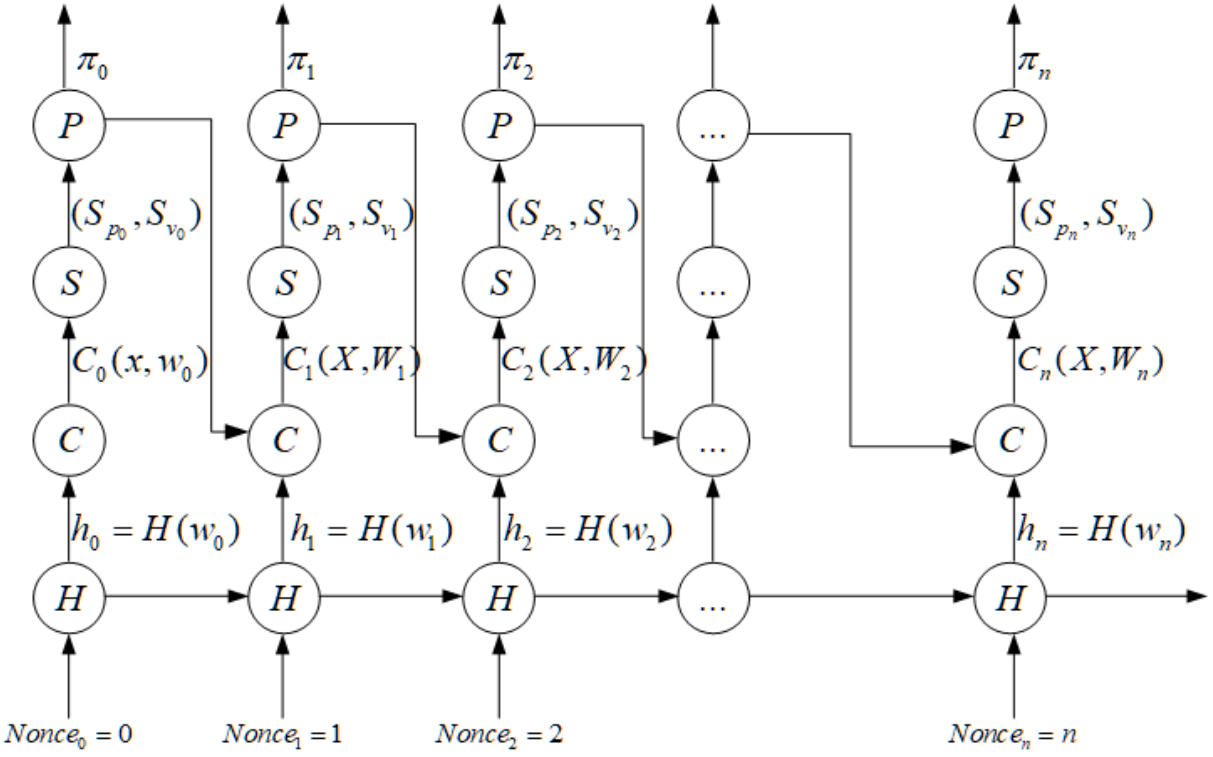

Рисунок Р.1 — Схема формування ланцюжка рекурсивних доказів обчислювальної цілісності

ДОДАТОК С

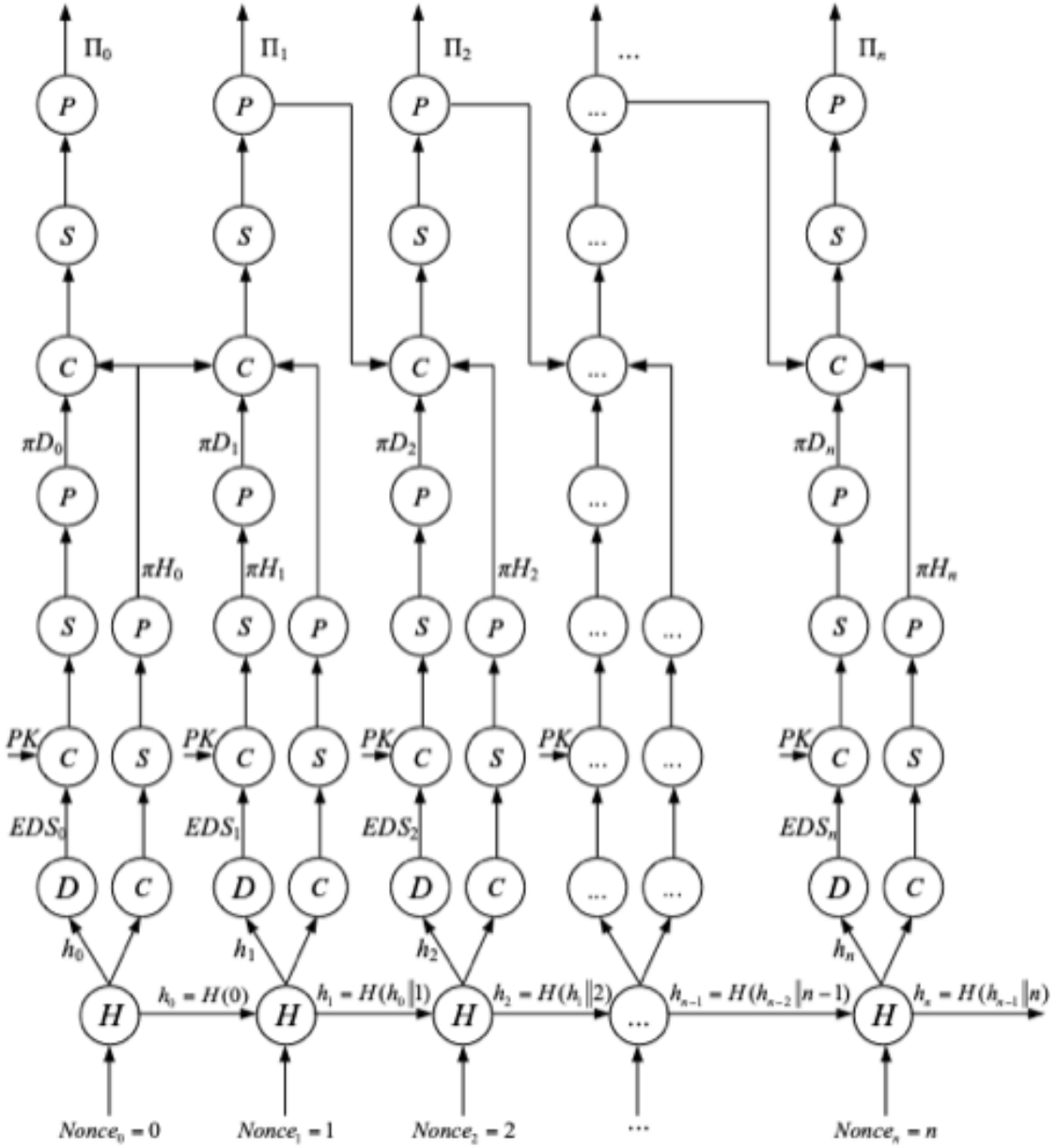

Рисунок С.1 — Схема формування ланцюжка доказів ОЦ з перевіркою ЦП

ДОДАТОК Т

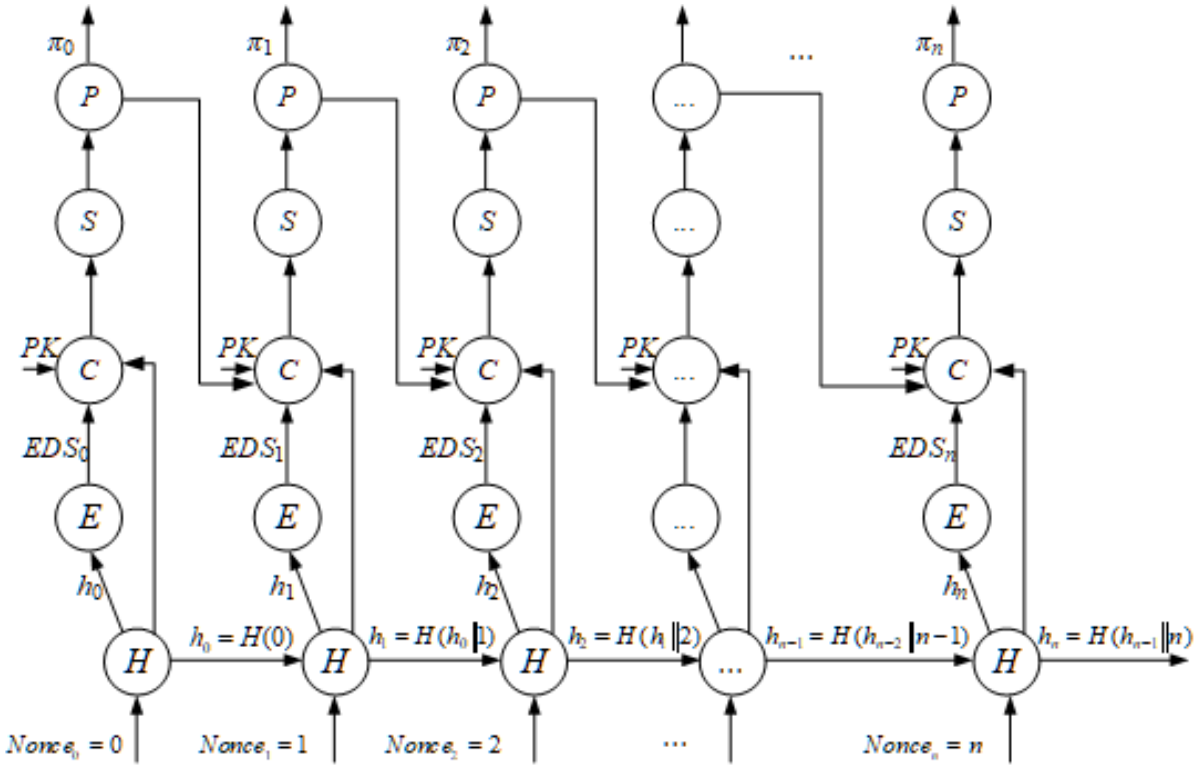

Рисунок Т.1 — Схема формування ланцюга доказів ОЦ з перевіркою ЦП

ДОДАТОК У

Таблиця У.1 – Доказ валідності хешу за алгоритмом SHA-256

```
pub fn sha256_proof<F: RichField + Extendable<D>, C:
GenericConfig<D, F = F>, const D: usize>(
    msg: &[u8],
    hash: &[u8],
) -> (CircuitData<F, C, D>, ProofWithPublicInputs<F, C, D>) {
    let len_in_bits = msg.len() * 8;
    let mut builder = CircuitBuilder::<F,
D>::new(CircuitConfig::standard_recursion_config());
    let targets = sha256_circuit(&mut builder, len_in_bits);
    let msg_bits = array_to_bits(msg);
    let hash_bits = array_to_bits(hash);
    let mut pw: PartialWitness<F> = PartialWitness::new();
    for i in 0..msg_bits.len() {
        pw.set_bool_target(targets.message[i], msg_bits[i]);
    }
    for i in 0..hash_bits.len() {
        pw.set_bool_target(targets.digest[i], hash_bits[i]);
        builder.register_public_input(targets.digest[i].target);
    }

    let timing = TimingTree::new("build", Level::Debug);
    let circuit_data = builder.build::<C>();
    timing.print();
    let timing = TimingTree::new("prove", Level::Debug);
    let proof = circuit_data.prove(pw).unwrap();
    timing.print();
    (circuit_data, proof)
}
```

Таблиця У.2 – Доказ валідності підпису за алгоритмом ED25519

```
pub fn ed25519_proof<F: RichField + Extendable<D>, C:
GenericConfig<D, F = F>, const D: usize>(
    msg: &[u8],
    sigv: &[u8],
    pkv: &[u8],
) -> (CircuitData<F, C, D>, ProofWithPublicInputs<F, C, D>) {
    let len_in_bits = msg.len() * 8;
    let mut builder = CircuitBuilder::<F,
D>::new(CircuitConfig::wide_ecc_config());
    let targets = ed25519_circuit(&mut builder, len_in_bits);
    let mut pw: PartialWitness<F> = PartialWitness::new();
    fill_ecdsa_targets::<F, D>(&mut pw, msg, sigv, pkv, &targets);
    let timing = TimingTree::new("build", Level::Debug);
    let circuit_data = builder.build::<C>();
    timing.print();
    let timing = TimingTree::new("prove", Level::Debug);
    let proof = circuit_data.prove(pw).unwrap();
    timing.print();
    (circuit_data, proof)
}
```

Таблиця У.3 – Перевірка доказів

```
let timing = TimingTree::new("verify", Level::Debug);
    let res = data.verify(proof);
    timing.print();
```

ДОДАТОК Ф

Таблиця Ф.1 — Час і результати вимірювань для ланцюжка доказів для хешів

№	Час для побудови схеми, с	Час генерації доказу, с	Розмір доказу, байти	Перевірка, с
0	0.529693	0.6954	132640	0.0104
1	1.0661488	1.7893	150268	0.0152
2	1.0135794	2.0026	150268	0.1044
3	1.0293587	2.4225	150268	0.0471
4	1.0987031	1.5323	150268	0.0957

Таблиця Ф.2 — Час і результати вимірювань для ланцюжка доказів для підписів

№	Час для побудови схеми, с	Час генерації доказу, с	Розмір доказу, байти	Перевірка, с
0	32.144226	72.0099	208376	0.0117
1	31.208908	94.6699	208376	0.01
2	29.096764	103.2076	208376	0.0105
3	30.003609	98.6119	208376	0.02
4	31.844381	69.1603	208376	0.0127

Таблиця Ф.3 — Час і результати вимірювань для ланцюга агрегованих доказів для підписів і хешів

№	Час для побудови схеми, с	Час генерації доказу, с	Розмір доказу, байти	Перевірка, с
0	1.3389456	1.6592	146348	0.0328
1	1.4501207	1.6903	146348	0.0358
2	1.5479413	1.8390	146348	0.0249
3	1.3872194	2.3278	146348	0.0251
4	1.3421979	1.6491	146348	0.0063

Таблиця Ф.4 — Кінцевий результат часу і результати вимірювань для ланцюжка доказів (агрегація)

№	Час для побудови схеми, с	Час генерації доказу, с	Розмір доказу, байти	Перевірка, с
0	34,0128646	74,3645	146348	0,0549
1	33,7251775	98,1495	146348	0,061
2	31,6582847	107,0492	146348	0,1398
3	32,4201871	103,3622	146348	0,0922
4	34,285282	72,3417	146348	0,1147

Таблиця Ф.5 — Час і результати вимірювань для ланцюжка рекурсивних доказів

№	Час для побудови схеми, с	Час генерації доказу, с	Розмір доказу, байти	Перевірка, с
0	31.563254	68.3191	211432	0.0274
1	29.565964	78.2291	211432	0.0174
2	29.53088	59.7240	211432	0.0192
3	30.960339	74.2933	211432	0.0382
4	32.037373	84.5603	211432	0.0279